Universidad Carlos III de Madrid
www.uc3m.es

TESIS DOCTORAL

The segmentation issue: general stopping criteria and specific design considerations for practical application of evolutionary algorithms

Autor:

José Luis Guerrero Madrid

Directores:

Antonio Berlanga de Jesús

José Manuel Molina López

Tutor:

Antonio Berlanga de Jesús

DEPARTAMENTO DE INFORMÁTICA

Colmenarejo, Marzo, 2015

Universidad Carlos III de Madrid
www.uc3m.es

TESIS DOCTORAL

THE SEGMENTATION ISSUE: GENERAL STOPPING CRITERIA AND SPECIFIC DESIGN CONSIDERATIONS FOR PRACTICAL APPLICATION OF EVOLUTIONARY ALGORITHMS

Autor: *José Luis Guerrero Madrid*

Directores: Antonio Berlanga de Jesús, José Manuel Molina López

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Secretario:

Calificación:

Colmenarejo, de de

Para mi familia, tanto la genética como la memética

Abstract

Segmentation is a tool presented for representation and approximation of data, according to a set of appropriate models. These procedures have applications to many different domains, such as time series analysis, polygonal approximation, Air Traffic Control,... Different heuristic and metaheuristic proposals have been introduced to deal with this issue. This thesis provides a novel multiobjective evolutionary method, analyzing the required general tools for the application evolutionary algorithms to real problems and the specific modifications required over the different steps of general proposals to adapt them to the segmentation domain.

An introduction to the domain is presented by means of the design of a specific heuristic for segmentation of Air Traffic Control (ATC) data. This domain has a series of characteristics which make it difficult to be faced with traditional techniques: noisy data and a large number of measurements. The proposal works on two phases, using a pre-segmentation which introduces available domain information and applying a standard technique over this initial technique's results. Its results according to the presented domain, tested with a set of eight different representative trajectories, show competitive advantages compared to general approaches, which oversegmentate noisy data and, in some cases, exhibit poor scalability. This heuristic proposal shows the costly process of adapting available approaches and designing specific ones, along with the multi-objective nature of the problem, which requires the use of quality indicators for a proper comparison process.

Applying evolutionary algorithms to segmentation provides several advantages, highlighting the fact that the problem dependance of heuristics make it costly to adapt these heuristics to new domains, as introduced by the designed heuristic to ATC. However, the practical application of these algorithms requires the study of a topic which has received little research effort from the community: stopping criteria. An evolutionary approach should contain a dynamic procedure which can determine when stagnation has taken place and stop the algorithm accordingly (as opposed to a-priori cost budgets, either in function evaluations or generations, which are usually applied for test datasets).

Stopping criteria have been faced for single and multi-objective cases in this thesis. Single-objective stopping criteria have been approached proposing an active role of the stopping criteria, actively increasing the diversity in the variable space while tracking the updates in the fitness function. Thus, the algorithm reuses the information obtained for the stopping decision and feeds it to a stopping prevention mechanism in order to prevent problematic situations such as early convergence. The presented algorithm has been tested according to a set of 27 different functions, with different characteristics regarding their dimensionality, search space, local minima... The results show that the introduced mechanisms enhance the robustness of the results, due to the improved exploration and the early convergence prevention.

Multi-objective stopping criteria are faced with the use of progress indicators (comparison measures of the quality of the evolution results at different generations) and an associated data gathering tool. The final proposal uses three different progress indicators, (hypervolume, epsilon and Mutual Dominance Rate) and considers them jointly according to a decision fusion

architecture. The stagnation analysis is based on the least squares regression parameters of the indicators values, including a normality analysis as well. The online nature of these algorithms is highlighted, preventing the recomputation of the indicators values which were present in other available alternatives, and also focusing on the simplicity of the final proposal, in order to reduce the cost of introducing it into available algorithms. The proposal has been tested with instances of the DTLZ algorithm family, obtaining satisfactory stops with a standard set of configuration values for the technique. However, there is a lack of quantitative measures to determine the objective quality of a stop and to properly compare its value to other alternatives.

The multi-objective nature of the segmentation problem is analyzed to propose a multi-objective evolutionary algorithm (MOEA) to deal with it. This nature is analyzed according to a selection of available approaches, highlighting the difficulties which had to be faced in the parameter configuration in order to guide the processes to the desired solution values. A multi-objective a-posteriori approach such as the one presented allows the decision maker to choose from the front of possible final solutions the one which suits him best, simplifying this process. The presented approach chooses SPEA2 as its underlying MOEA, analyzing different representation and initialization proposals. The results have been validated against a representative set of heuristic and metaheuristic techniques, using three widely extended curves from the polygonal approximation domain (chromosome, leaf and semicircle), obtaining statistically better results for almost all the different test cases.

This initial MOEA approach had unresolved issues, such as the archiving technique complexity order, and also lacked the proper specific design considerations to adapt it to the application domain. These issues have been faced according to different improvements. First of all, an alternative representation is proposed, including partial fitness information and associated fitness-aware transformation operators (transformation operators which compute children fitness values according to their changes and the parents partial values). A novel archiving procedure is introduced according to the bi-objective nature of the domain, being one of them discrete. This leads to a relaxed Pareto dominance check, named epsilon glitches.

Multi-objective local search versions of the traditional algorithms are proposed and tested for the initialization of the algorithm, along with the stopping criterion proposal which has also been adapted to the problem characteristics. The archive size in this case is big enough to contain all the different individuals in the optimal front, such that quality assessment is simplified and a simpler mechanism can be introduced to detect stagnation, according to the improvements in each of the possible individuals.

The final evolutionary proposal is scalable, requires few configuration parameters and introduces an efficient dynamic stopping criterion. Its results have been tested against the original technique and the set of heuristic and metaheuristic techniques previously used, including the three original curves and also more complex versions of them (obtained with an introduced generation mechanism according to these original shapes). Even though the stopping results are very satisfactory, the obtained results are slightly worse than the original MOEA for the three simpler problem instances with the established configuration parameters (as was expected, due to the computational effort of the a-priori established number of generations and population size, based on the analysis of the algorithm's results). However, the comparison versus the alternative techniques stills shows the same statistically better results, and its reduced computational cost allows its application to a wider set of problems.

Resumen

La segmentación es una técnica creada para la representación y la aproximación de conjuntos de datos a través de un conjunto de modelos apropiados. Estos procedimientos tienen aplicaciones para múltiples dominios distintos, como el análisis de series temporales, la aproximación poligonal o el Control de Tráfico Aéreo. Se han hecho múltiples propuestas tanto de carácter heurístico como metaheurístico para lidiar con este problema. Esta tesis proporciona un nuevo método evolutivo multiobjetivo, analizando las herramientas generales necesarias para la aplicación de algoritmos evolutivos a problemas reales y las modificaciones específicas necesarias sobre los distintos pasos de las propuestas genéricas para adaptarlos al dominio de la segmentación.

Se presenta una introducción al dominio mediante el diseño de una heurística específica para la segmentación de datos procedentes del Control de Tráfico Aéreo (CTA). Este dominio tiene una serie de características que dificultan la aplicación de técnicas tradicionales: datos con ruido y un gran número de muestras. La propuesta realizada funciona de acuerdo a dos fases, utilizando una presegmentación que introduce información del dominio disponible para posteriormente aplicar una técnica estándar sobre los resultados de esta técnica inicial. Sus resultados para el dominio presentado, probado con un conjunto de ocho trayectorias representativas distintas, presentan ventajas competitivas frente a los enfoques generales, que sobresegmentan los datos con ruido y, en algunos casos, presentan una mala escalabilidad. Esta propuesta heurística muestra el costoso proceso que implica adaptar los enfoques existentes o el diseño de otros nuevos, junto a la naturaleza multiobjectivo del problema, que precisa del uso de indicadores de calidad para realizar un proceso de comparación apropiado.

La aplicación de algoritmos evolutivos a la segmentación tiene múltiples ventajas, destacando el hecho de la dependencia existente entre las heurísticas y el problema específico para el que han sido diseñadas, lo que hace que su adaptación a nuevos dominios sea costosa, como se ha introducido a través de la propuesta heurística para CTA. A pesar de ello, la aplicación práctica de estos algoritmos requiere el estudio de una faceta que ha recibido poca atención por parte de la comunidad desde el punto de vista de la investigación: los criterios de parada. Un enfoque evolutivo debería tener una técnica dinámica que pueda detectar cuando se ha producido el estancamiento del proceso, y parar el algoritmo de acuerdo a ello (de manera opuesta a los criterios a-priori que establecen un coste predeterminado, expresado como número de evaluaciones o de generaciones, y que son habitualmente aplicados para los conjuntos de datos de prueba)

Los criterios de parada se han afrontado tanto desde el caso de un único objetivo como desde el caso multiobjectivo en esta tesis. Los criterios de parada para un único objetivo se han abordado proponiendo un rol activo para el criterio, aumentando la diversidad en el espacio de variables de una manera activa, mientras se monitorizan los cambios en la función objetivo. De esta manera, el algoritmo reutiliza la información obtenida para la decisión de parada y la inserta en un mecanismo de prevención de la parada con la finalidad de prevenir situaciones problemáticas como la convergencia temprana. El algoritmo presentado se ha probado sobre un conjunto de 27 funciones distintas, con diferentes características respecto a su dimensionalidad, espacio de búsqueda, mínimos locales... Los resultados muestran

que los mecanismos introducidos mejoran la robustez de los resultados, haciendo uso de la exploración mejorada y la prevención de la convergencia temprana.

Los criterios de parada multiobjetivo se han planteado con el uso de indicadores de avance (medidas comparativas de la calidad de los resultados de la evolución en diferentes generaciones) y una herramienta de recolección de datos asociada. La propuesta final utiliza tres indicadores de avance distintos (hypervolumen, epsilon y ratio de dominancia mutua) y los considera de una manera conjunta de acuerdo a una arquitectura de fusión de decisiones. El análisis del estancamiento se basa en los parámetros de una regresión de mínimos cuadrados sobre los valores de los indicadores, incluyendo así mismo un análisis de normalidad. Se recalca la naturaleza online de estos algoritmos, evitando el recálculo de los valores de los indicadores que estaba presente en otras alternativas disponibles, y también focalizándose en la simplicidad de la propuesta final, de manera que se facilite el proceso de introducir el criterio en los algoritmos existentes. La propuesta ha sido probada con instancias de la familia de algoritmos DTLZ, obteniendo resultados de parada satisfactorios con un conjunto de valores de configuración estándar para la técnica. Sin embargo, existe una falta de medidas cuantitativas para determinar la calidad objetiva de una parada, así como para comparar de manera apropiada su valor frente al de otras alternativas.

La naturaleza multiobjetivo del problema de segmentación se ha analizado para proponer un algoritmo evolutivo multiobjetivo (AEMO) para resolverlo. Esta naturaleza ha sido analizada de acuerdo a una selección de los enfoques disponibles, destacando las dificultades que se tienen que afrontar en la configuración de los parámetros de cara a guiar el proceso hacia los valores de solución deseados. Un enfoque multiobjetivo a-posteriori como el que se ha presentado permite al responsable elegir del frente de posibles soluciones finales aquella que encaja mejor, simplificando este proceso. El enfoque presentado ha elegido SPEA2 como algoritmo de base, analizando diferentes propuestas de inicialización y representación. Los resultados se han validado frente a un conjunto significativo de técnicas heurísticas y metaheurísticas, utilizando tres curvas ampliamente extendidas en el dominio de la segmentación poligonal (cromosoma, hoja y semicírculo), obteniendo resultados estadísticamente mejores para la casi totatilidad de los casos de prueba.

Esta propuesta inicial de AEMO presentaba una serie de problemas sin resolver, como el orden de complejidad de la técnica de almacenaje, y además carecía de las consideraciones específicas de diseño para su adaptación al dominio de aplicación. Estos problemas se han afrontado de acuerdo a diferentes mejoras. Por un lado, se ha propuesto una representación alternativa, incluyendo información parcial de la función objetivo y operadores de transformación informados (operadores de transformación que calculan los valores de la función objetivo de los hijos de acuerdo a los cambios realizados y los valores parciales de los padres). Una nueva técnica de almacenaje se ha introducido de acuerdo a la naturaleza biobjetivo del dominio, siendo uno de ellos además discreto. Esta naturaleza ha llevado a la aplicación de una forma relajada de dominancia de Pareto, que hemos denominado pulsos épsilon.

Versiones multiobjetivo de los algoritmos tradicionales de búsqueda local han sido propuestas y probadas para la inicialización del algoritmo, junto con la propuesta de criterio de parada, que también ha sido adaptada a las características del problema. En este caso, el tamaño del almacén es suficientemente grande como para almacenar todos los individuos del frente óptimo, de manera que las técnicas de análisis de calidad de los frentes se simplifican, y un mecanismo más sencillo puede ser introducido para detectar el estancamiento, de acuerdo

a las mejoras en cada uno de los individuos posibles.

La propuesta evolutiva final es escalable, requiere pocos parámetros de configuración e introduce un criterio de parada dinámico y eficiente. Sus resultados se han probado frente a la técnica original y el conjunto de técnicas heurísticas y metaheurísticas previamente utilizadas, incluyendo las tres curvas originales y versiones más complejas de las mismas (obtenidas con un mecanismo de generación incluido de acuerdo a estas tres formas originales). A pesar de que los resultados de parada son muy satisfactorios, los resultados obtenidos son ligeramente peores que el AEMO original para las tres instancias del problema más simples, utilizando el conjunto de parámetros de configuración establecidos (como cabía esperar, dado el coste computacional del número de generaciones y tamaño de la población establecidos a priori, basados en el análisis de los resultados del algoritmo). En cualquier caso, la comparación frente a las técnicas alternativas todavía presenta los mismos resultados estadísticamente mejores, y las mejoras en el coste computacional permiten su aplicación a un mayor conjunto de problemas.

Agradecimientos

> *Existen infinitas posibilidades de realidad alternativas Sólo que para ti es imposible verlas Tus sentidos no están preparados para verlas*
>
> Carlos Escudero, *¡Permítete ser feliz!*, 2006

Espero que aquellos que estén interesados en esta tesis y no hablen español me perdonen por esta sección inicial, pero algunas de las personas que aparecen en ella no hablan inglés, y ya que han tenido que soportarme hablando demasiado a menudo sobre cosas relacionadas con esta tesis que ni entendían ni probablemente les importaban lo más mínimo, creo que merecen este pequeño y a todas luces insuficiente homenaje.

Ahora que este proceso para mí está acabando y que he compartido este camino con bastantes personas en una situación parecida, creo que, para todo el mundo, hacer una tesis es un esfuerzo enorme que no se lograría sin el apoyo de mucha gente. Para muchos de nosotros es nuestro primer gran proyecto a largo plazo (supongo que si obviamos la vida en sí misma) y eso causa vértigo (casi como mirar al abismo a los ojos... pero no quiero insertar citas encubiertas). En mi caso esto es cierto en un grado todavía mayor del habitual por las circunstancias especiales que han rodeado mi vida a lo largo de estos años, por lo que esta sección es incluso más importante de lo habitual.

Tengo que empezar por las dos personas que figuran en el título de este tomo junto al mío, mis directores José Manuel Molina y Antonio Berlanga. A ambos les conocí como profesores míos en la carrera, y el paso de este tiempo sólo ha conseguido que crezcan el aprecio que les tengo como personas y el respeto que como profesionales siento por ellos. Sé que a veces dirigirme no ha sido fácil, y os estoy muy agradecido por todo lo que habéis hecho por mí.

El siguiente paso son las personas que profesional y personalmente han compartido momentos y colaborado conmigo a lo largo de estos años. Aquí tengo que incluir al resto de profesores titulares de mi grupo de investigación, Jesús García, Javier Carbó y Miguel Ángel Patricio, a aquellos que estuvieron al menos un tiempo con nosotros, como Juan y David, a mis compañeros de laboratorio a lo largo del doctorado, Álvaro, Quique, Gonzalo, Alberto y Miguel, y a los que terminaron antes que nosotros y nos marcaron el camino a seguir, Luis, Nayat y Fede. Una mención especial tiene que tener también Mar, por un lado compañera y por otro la persona a la que íbamos siempre con los líos de congresos y trámites en general, y que lidiaba con la burocracia de la universidad en nuestro nombre (y siempre te recibía con una sonrisa y una palabra amable). Podría hacer un recuento individual de las muchas cosas que he aprendido de cada uno de vosotros, pero sobre todo habéis hecho que las partes oscuras de este camino parecieran un poco menos impenetrables.

Profesionalmente podría citar a muchas más personas con las que, de una manera u otra, he coincidido y colaborado a lo largo de estos años, pero estoy seguro que sería una lista parcial e incompleta, por lo que prefiero obviarla. Sin embargo, han estado ahí, comentando

en los congresos, haciendo sugerencias, dando tutoriales inspiradores o colaborando en cursos y proyectos. Siempre que te encuentras cara a cara con alguien que para ti hasta ese momento era un gran apellido al inicio de tus citas sientes una sensación muy particular que humaniza en gran parte el mundo de la investigación.

Estos años, además de investigación, han sido para mí años de docencia, donde he encontrado mi auténtica vocación desconocida. He vivido muchas cosas también como profesor, la inmensa mayoría de ellas muy buenas (a pesar de los hitos desagradables que no pueden evitarse, supongo que como en todas las facetas de la vida). Los dos profesores principales con los que compartido aula, Félix y Cristina, me han facilitado mucho todo el proceso con su experiencia y sus consejos. Creo que algo he conseguido transmitir en esas clases, como mínimo espero que mi pasión por lo que hacía, pero de lo que estoy seguro es de que he recibido muchísimas cosas de mis alumnos: vitalidad, ilusión, esperanza, e infinitas lecciones vitales más, tanto personales como profesionales. Dando clase te das cuenta de que todos necesitamos expandir nuestras miras. De nuevo un agradecimiento especial para Tomás y Jacobo (una de las personas con las que más me he reído en mi vida), cuyos proyectos fin de carrera y fin de grado, respectivamente, he dirigido y disfrutado.

Y llegamos al terreno más personal, empezando por mi familia. Como me recordó Carlos, con nuestros padres tenemos una deuda perpetua que nunca podremos saldar, ya que nos han dado la vida. Con toda seguridad, ellos son los que más han sufrido con todos y cada uno de mis avatares, los que me han acompañado en cada momento y luchado sistemáticamente a mi lado y los que más se han regocijado en mis victorias. Una de las cosas que estos años me han recordado es la importancia de sentirse agradecido con aquellos que lo merecen. Y nadie lo merece más que vosotros. Papá, mamá, os quiero mucho.

Mi familia tiene muchas más personas extraordinarias. En el terreno más cercano, dos hermanas con dos niños cada una dan para mucho. A algunos les veo menos de lo que me gustaría, pero son los caminos de la vida. Al menos tengo la suerte de poder disfrutar con los que sí comparto mi día a día. Hasta tengo un cuñado genial (no sé si está permitido mostrar agradecimiento público hacia un cuñado, pero me voy a arriesgar) que me ayudó cuando tuve que hacer rehabilitación con bicicleta después de una rotura. Y por supuesto, Jorge y Alejandro, mis sobrinos que viven en Madrid, que me siguen queriendo a pesar de que ya no soy "el mejor jugador de videojuegos del mundo", y son dos de las personas que yo más quiero en el universo.

He tenido que cambiar en estos años mis aficiones y la manera en la que pasaba mi tiempo libre por determinadas circunstancias, y eso ha hecho, en parte, que el resto del entorno de mi vida cambiara junto con ellas. Hay personas que siempre han estado a mi lado a las que les debo mucho de este trabajo y en general de mi vida, especialmente a Laura, sin ella no tendría muchas de las cosas buenas que tengo y probablemente sí todas las malas. Muchas otras personas han surgido en mi vida a raíz de estos cambios, y me han ayudado más de los que ellas mismas piensan.

La comunidad de patinadores de Madrid son un grupo de gente increíble, con las que me lo he pasado fenomenal, y que me han empujado a formar esas aficiones nuevas para mi vida. Un agradecimiento especial a mis profesores en Tres60, Sara y Nacho, que son un sol (todavía tengo la imagen de Nacho sosteniéndome de las manos el día de la rotura con un cigarrillo perenne en la boca). No voy a ponerme a hacer una lista de despilfarradores porque sería eterna, ellos saben quiénes son, espero (al fin y al cabo, en esta época de prisas y falta de tiempo constante, la mejor manera de demostrarle a alguien tu aprecio es dedicarle tu

tiempo) y con eso me basta. No puedo olvidar especialmente el apoyo de Chus, una de las personas más sabias y dulces con las que he tenido la suerte cruzarme en la vida.

Merecen su hueco propio en esta sección mis venerados Cullerots, compañeros de tantos y tantos veranos y momentos inolvidables, que todavían se encargan de recordarme todos los motes que he tenido en mi vida, de reírse de mis chorradas, de seguir pidiéndome la letra de "deseo, anhelo" y, en general, de hacerme cobrar una perspectiva que a veces pierdes al verte sumergido en tus problemas diarios. Incluso vienen a darte con un periódico en el hocico si te portas mal, como Laura y Borja. Gracias, de verdad.

Por último, un pensamiento para los que ya no están, pero que son responsables también de esto. No puedo dejar de recordar los domingos por la mañana paseando con mi tío (una de las personas más cultas que jamás he conocido) por la Cuesta Moyano, descubriendo el placer de conocer cosas nuevas. Os echo mucho de menos, siempre.

Contents

List of Figures

List of Tables

1

Introduction

> *'Begin at the beginning', the King said, gravely, 'and go on till you come to an end; then stop'*

Lewis Carroll, *Alice in Wonderland*, 1899

1.1 General introduction

Segmentation is a tool presented for representation and approximation of data, according to a set of appropriate models. These models may vary among Fourier transforms, wavelets, linear models or multi-model approaches, among others. The roots of this domain can be traced back to the fifties, related to the studies on human processing and understanding of visual information. This processing does not only reduce the required amount of data for the representation of the information, but also makes possible the application of additional algorithms related to the comparison and analysis of that information, such as feature extraction, or provides valuable information in its processed information by itself. This kind of information analysis plays a crucial role in the knowledge society. Segmentation processes are applied to different domains, such as Air Traffic Control, time series analysis or polygonal approximation

The original objective when facing a new problem is to obtain the optimal solution for that problem (exact algorithms). Unfortunately, this cannot be performed for a high number of different problems, due to their inherent difficulty (in terms of computational complexity). The next best alternative would be to provide a solution with a bounded distance to the optimal one (approximation algorithms). Again, this may not be feasible for particularly difficult problems (which is a category many real problems tend to fall into). When none of the two previous alternatives is feasible, the researcher / practitioner has to resort to *best effort* algorithms, where a solution is obtained without proper information about its quality (heuristic approaches).

Problems are classified according to their difficulty (such as P and NP-hard problem classifications) and the required time to solve them. NP-hard problems, for instance, require exponential time to be solved by a deterministic algorithm (their proper definition will be provided in section 2.2), and thus they usually require heuristic resolution methods. One of the issues related to heuristic approaches is that their extreme problem-dependence, such that

for every new faced problem, a new solution has to be built from scratch. This issue has been faced with metaheuristics, which can be seen as general application guides or frameworks for specific problem heuristics.

According to their objectives, metaheuristics have to deal with two different processes: the exploration of the search space and the exploitation of the information which has already been acquired. Different metaheuristic approaches have different focuses on each of these processes, and also different combinations of them have been proposed to try to combine in the best possible way these two requirements (such as memetic algorithms or hyperheuristics).

Evolutionary algorithms are population based metaheuristics, meaning that a whole population of individuals is evolved at every step, instead of a single solution. With some of the Darwinism ideas at their core, evolutionary algorithms apply crossover and mutation operators over a population of solutions, preserving a certain number of them every generation, until the established stopping criterion is met. Different representations have been brought into this schema with remarkable results, from the original strings of 0's and 1's which were used for genetic algorithms up to the sparse trees used in genetic programming. Many real life problems have been properly tackled with these approaches as well (taking advantage of the ever growing computational power of computers and the use of parallel programming techniques, which are inherent to population based metaheuristics).

Additional issues arise when the optimization problem requires not only one, but several objectives in conflict which have to be optimized jointly. Again, real life examples falling into this category are easy to find, such as obtaining the maximum profit minimizing the investment for a given financial operation. This category of problems is called Multi-objective problems (MOPs). Evolutionary algorithms have been studied and applied into this particular category of problems, being usually known as Multi-objective evolutionary algorithms (MOEAs). As the number of objectives is increased (Multi-objective optimization is usually focused on two objectives), the optimization process performed by the algorithms dealing with them becomes known as Many-objective optimization (presenting specific issues and approaches).

The design and validation of metaheuristics present, due to their stochastic nature, a very particular issue: the quality assessment of newly devised approaches. This problem reaches higher complexity levels for algorithms with more than a single fitness function being optimized. Different approaches have been presented, based on concepts such as the statistical properties of the population (either in fitness space of the one presented by the variables being codified) or the so called *quality indicators*, which reduce the dimensionality of multi-objective problems to ease the difficulty in the assessment function.

Research studies have faced many of the properties, characteristics and limitations of evolutionary algorithms. However, the study of the convergence issue (particularly regarding that final step of the required stopping criterion) has been usually neglected. The need for well-established stopping criteria is crucial for the application of evolutionary algorithms to real problems. This topic has been traditionally faced establishing an *a-priori* budget (either in computation time, evaluations or evolution generations) or being directly supervised by the decision maker (expert) in the problem.

Thus, stopping criteria are a required general application tool for evolutionary approaches in order to apply them to real problems. Regarding these tools, the idea behind this work is the close relationship which exists between quality assessment and stopping criteria. Quality assessment was based on the comparison of a series of solutions from different runs of a

series of evolutionary algorithms, comparing the results (regardless of whether these results are single solutions or sets of them including different fitness functions values). A possible stopping criteria approach would be to compare the evolution at different steps (or step by step at each generation) and determine whether the evolution has converged according to that information. This would reapply the techniques designed for quality assessment, modifying them accordingly to the requirements of their new application issue (basically comparing different populations from a single run of an evolutionary algorithm).

This approach arises several issues: the information required for this comparison, how this information is gathered and the techniques required for its comparison, the differences in the application in single and multi-objective optimization... This work will focus first on single-objective stopping criteria, considered as a simpler problem, in order to face multi-objective issues as the following step towards a general approach to the convergence issue. In fact, the use of quality indicators reduces multi-objective problems to a single value in terms of comparison, suggesting that the chosen approach could lead to a valid general thesis schema.

Returning to the application domain, many different heuristic approaches have been developed and presented for segmentation. As previously introduced, the main handicap of heuristics are their problem dependence. This has led to the development of similar techniques in the different domains which segmentation is applied to. Also, the redesign of heuristics for specific problem characteristics is costly and inefficient. An initial heuristic approach to the characteristics of the Air Traffic Control domain will be introduced in order to analyze this issue in depth, establish the required solution characteristics, and lead the thesis development, guided by these characteristics. The final proposal of current work is to develop a metaheuristic approach in order to cope with segmentation domain requirements in a more appropriate way.

1.2 Objectives

To achieve the presented task, this work will be focused on the following individual objectives.

- Describe and analyze relevant state of the art for the thesis: metaheuristics, evolutionary computing and segmentation
- Analyze of multi-objective problems and their related procedures: approaches to deal with several objective functions jointly, Pareto dominance, diversity management, quality assessment...
- Overview general stopping criteria present in the literature, both for multi and single objective approaches.
- Present a specific heuristic approach for the segmentation domain based on Air Traffic Control data.
- Propose a single objective stopping criterion tool for evolutionary algorithms.
- Propose a multiobjective stopping criterion tool for evolutionary algorithms, including the information source and the data gathering performed in order to determine the stopping decision.

- Propose a general metaheuristic approach based on evolutionary algorithms to segmentation, along with the required tools for proper comparison to some of the available heuristic approaches.
- Propose a final metaheuristic approach, modifying its different operators in order to adapt them to the application domain as required.

1.3 Document structure

Each of the remaining chapters of this work presents a short guidance into the general objectives of the work, along with individual introductions, conclusions and future lines for the topics dealt within each individual chapter. There are three exceptions to this structure: chapter 2, fundamentals, which lacks an introductory section (being this role fulfilled by current chapter), chapter A, the final appendix, which introduces a set of unconstrained single objective optimization functions, being an extension to the proposal presented in chapter 4, and thus providing no introduction or conclusion sections, and finally chapter 7, which revisits the general proposal of chapter 6, and, thus, has no individual introduction. The chapters are arranged according to the following structure:

Chapter 2: Fundamentals This chapter will deal with the presentation of the most important fundamentals for the thesis proposal. This presentation is focused on the main topics required for the different proposals: nature of the problems treated, evolutionary algorithms and their different operators, multi-objective problems, stopping criteria (both for single and multi-objective evolutionary algorithms), quality assessment and finally the segmentation issue, covering the formalization and general design of available techniques.

Chapter 3: An initial non-evolutionary approach to the application domain: HLRA This chapter presents an initial approach to the segmentation issue using data coming from the Air Traffic Control domain. This approach will highlight the required complex changes to adapt segmentation heuristics among different domains, along with a special focus on comparison issues. These issues lead to the use of quality indicators to perform these comparisons, and provide the initial lead to the multi-objective nature of segmentation processes.

Chapter 4 Single-objective stopping criteria for evolutionary algorithms This chapter presents a stopping criterion proposal for single-objective optimization. This proposal is based on two steps: the first approximation covers a series of modifications over Evolutionary Strategies Learned with Automated Termination criterion (ESLAT) algorithm, leading to the introduction of the Robust ESLAT algorithm (R-ESLAT). Special attention will be paid to the incorporated stopping criterion and its modifications, according to the approaches highlighted in the state of the art. After that, the stopping criterion is proposed following an active approach, which means that the criterion modifies the population characteristics to prevent early convergence, measuring, at the same time, when this convergence has taken place.

Chapter 5: Multi-objective stopping criteria for evolutionary algorithms This chapter presents two approaches to multi-objective stopping criteria based on the use of quality

indicators. The first proposal introduces the use of a Kalman estimation based technique with a fusion architecture to handle multiple quality indicators jointly providing enhanced robustness in the final criterion. The second alternative is focused on simplicity and efficiency, attempting to provide an easily implementable technique to be introduced in available MOEAs. LSSC proposal is focused on least squares estimation including two different stopping tests, one regarding the statistical representativeness of the linear estimation and a second one guided by a configuration parameter determining the minimal improvement per generation before the algorithm is stopped.

Chapter 6: Multiobjective evolutionary polygonal approximation This chapter presents an initial evolutionary proposal for the segmentation issue. The choice is presented according to an analysis of available techniques, detailing their multi-objective handling techniques according to the theoretical alternatives, and leading to an a-posteriori MOEA proposal. This proposal includes the selection of the required representation, underlying MOEA algorithm and the configuration of several initialization alternatives, focusing on the representativeness of diversity in the objective and variable spaces and the possible use of local search techniques.

Chapter 7: An efficient approach to multiobjetive evolutionary polygonal approximation This chapter analyzes in depth each of the steps of the general MOEA introduced in chapter 6, from the representation to the stopping criterion, and introduces the required modifications to enhance their efficiency for the application domain. This chapter also highlights the difficulties of the application of general algorithms to problems with specific characteristics, and how adapting them to these characteristics may improve their performance.

Chapter 8: Conclusions and future lines . This chapter summarizes the results presented according to the objectives set in current chapter. Along with these results, the opened research paths for the studied topics will be presented, concluding this thesis.

Appendix: Single Optimization Function Set . Chapter 4 requires the use of a certain function set in order to perform the required performance comparison between the presented approach and a representative alternative from the state of the art. This function set is overviewed within the chapter, but not completely described. This appendix will present this description, including dimensionality, search space, formulation and bidimensional representation.

1.4 Visual overview

As presented, the different chapters of this thesis will be focused on different aspects related to the objectives, even introducing their own focused states of the art and conclusions. In order to enhance the readability of the document and facilitate the understanding of the relationships between the different sections of it, figure 1.1 shows a visual overview of the domain and the different phases established to deal with the introduced issues detected.

Figure 1.1: Visual overview of the different phases of the research and their relationship towards the final proposal

2

Fundamentals

> *'That's right', shouted Vroomfondel, 'we demand rigidly defined areas of doubt and uncertainty!'*
>
> Douglas Adams, *The HitchHiker's Guide to the Galaxy*, 1979

2.1 Optimization methods

In an ideal world, any problem would be solved obtaining its optimal solution, or at least, a solution which has a known (or boundary restricted) distance to its optimal one (in order to allow the researcher to measure its quality). Those algorithms are known, respectively, as exact or approximation algorithms.

Among the most important exact algorithms are: brand and X family (branch and bound, branch and cut and branch and price), constraint programming, dynamic programming and A* search algorithms (the individual details for each of these techniques may be consulted in (Russell et al., 1995). The general idea for exact methods is to cover the whole interest zones of the search space, subdividing it into simpler problems.

Approximation algorithms, on the other hand, do not provide the optimal solution for a given problem, but they guarantee a certain quality bound with respect to the global optimum (and also within certain run time boundaries) (Hochba, 1997). Among them, ϵ-approximation algorithms may be highlighted (where an obtained solution x is not worse than ϵ times the optimal solution s) (Vazirani, 2001).

However, many important problems, due to their complexity, cannot be solved using any of those solution approaches. Then, researchers may resort to heuristic approaches, in order to simplify their problems and obtain solutions for them. These algorithms perform a *best effort* approach, trying to obtain good solutions (where this quality may be difficult to measure, since its distance to the optimal one is unknown). Heuristic algorithms, along with approximation ones, are known as approximate algorithms. This difficulty to measure the quality of the obtained solutions, which is intrinsically linked with the use of approximate methods, will be repeatedly highlighted through this work, due to the issues that this fact introduces for some of the approaches presented in this work.

A colloquial definition of heuristic, which could be provided by many computer science grade students, would define a heuristic as a *trick* to reduce the complexity of a hard problem,

being this easier problem the one which is solved, and using this solution in order to resolve the original one. This definition is really close to the formal one, and is, at the same time, probably more explicative. Heuristic comes from the Greek word *heuriskein*, which means the art of discovering new strategies to solve problems.

Heuristics are very problem specific, with a resulting poor reusability. This leads to the development of higher level heuristics, usually called metaheuristics (a term originally introduced in (Glover, 1986)). The *meta* prefix introduces the meaning of a higher level. In practice, metaheuristics may be seen as templates where the problem specific heuristics may be inserted (but the template itself can be reused and applied to a wide range of different problems).

This generalization process towards reusability can be found in the different disciplines of computer science. Due to its more concrete nature, it may be especially visible in the evolution of the different programming paradigms, where the change from procedural to object oriented approaches was one of these attempts of generalization towards reusability (a general overview over the different paradigms and evolutions between them may be looked up in (Van-Roy & Haridi, 2004)). But this effort is a continued one (where, for instance, C++ language templates (Stroustrup, 1997) or agent-oriented paradigms (Huntbach & Ringwood, 1999)are additional attempts to increase the abstraction level).

In a similar way, metaheuristics are not the highest abstraction level for optimization problems. Considering metaheuristics as general frameworks for the application of particular heuristics, we might define general frameworks for the application and combination of metaheuristics themselves. First steps towards this directing were achieved by means of memetic algorithms (Moscato, 2000) (which combined particular metaheuristics in a very specific way), introducing hybrid metaheuristics. The definitive generalization step was taken by hyperheuristics (Burke et al., 2003): general frameworks for the application of particular metaheuristics capable of resolving general classes of problems. Figure 2.1 shows an overview of the different introduced optimization methods.

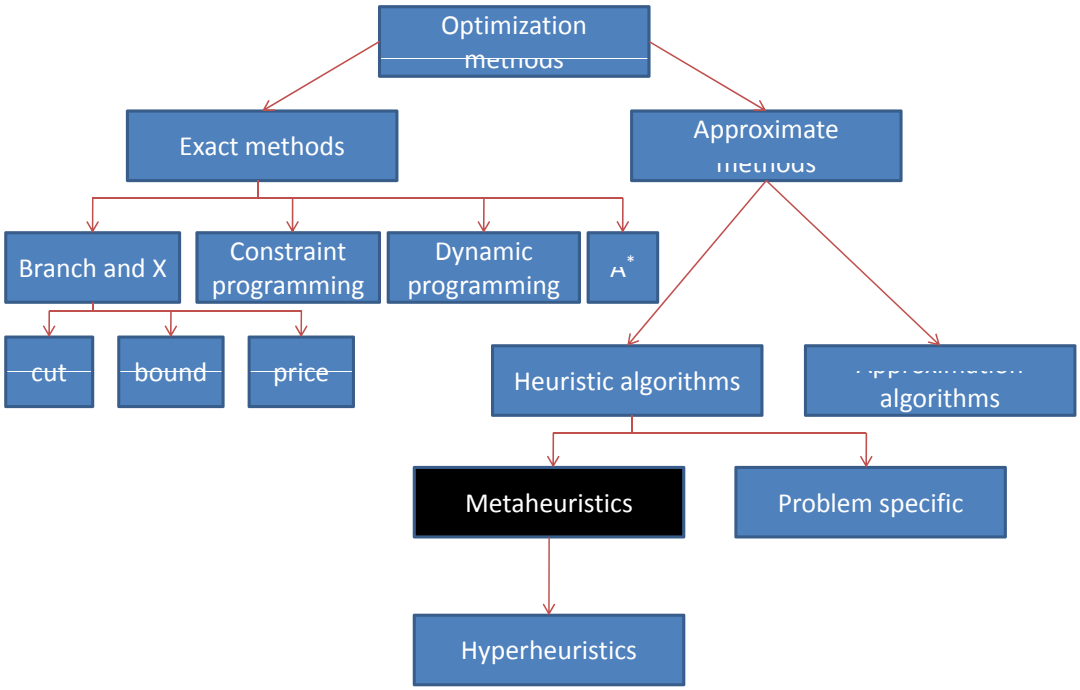
This work will be focused on evolutionary algorithms (Bäck, 1996), a particular instance of metaheuristics, which we will describe in detail in section 2.4.

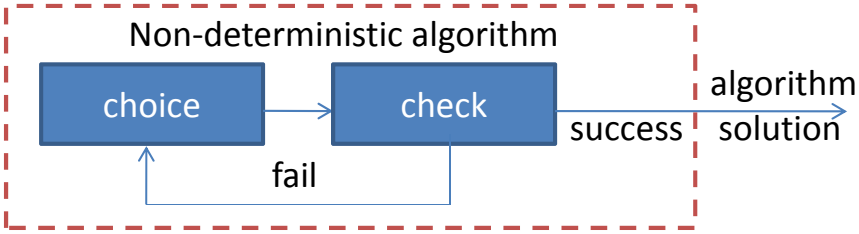
2.2 Problem complexity and categorization

In the literature it is usual to find statements like the following: metaheuristics are the appropriate method to deal with hard problems. The definition of easy and hard problems in this domain may be easily confused with their more usual meaning (which is fuzzy in its nature) so we would like to fix those meanings and, at the same time, introduce the need for the application of metaheuristics in a more concrete context.

A problem is easy (or tractable) if there exists a polynomial-time algorithm to solve it, whereas a problem is hard (or intractable) if no such algorithm exists. Considering these definitions, the introduced statement regarding the application of metaheuristics to hard problems is self-evident: easy problems should be solved with their associated polynomial time algorithm. It is also interesting to notice that the easy / hard attribute of a problem does not depend intrinsically on the problem's characteristics (although it is be derived from them), but on the solution found to deal with it.

The complexity theory deals with decision problems (yes or no answer), but is applicable to any optimization problem, since any of them can always be reduced to a decision one.



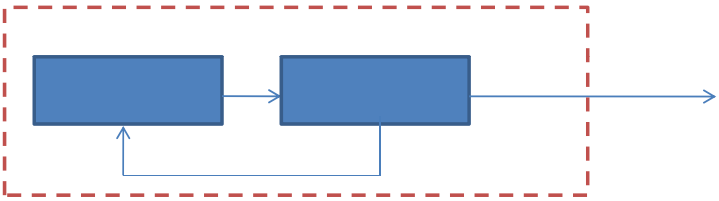


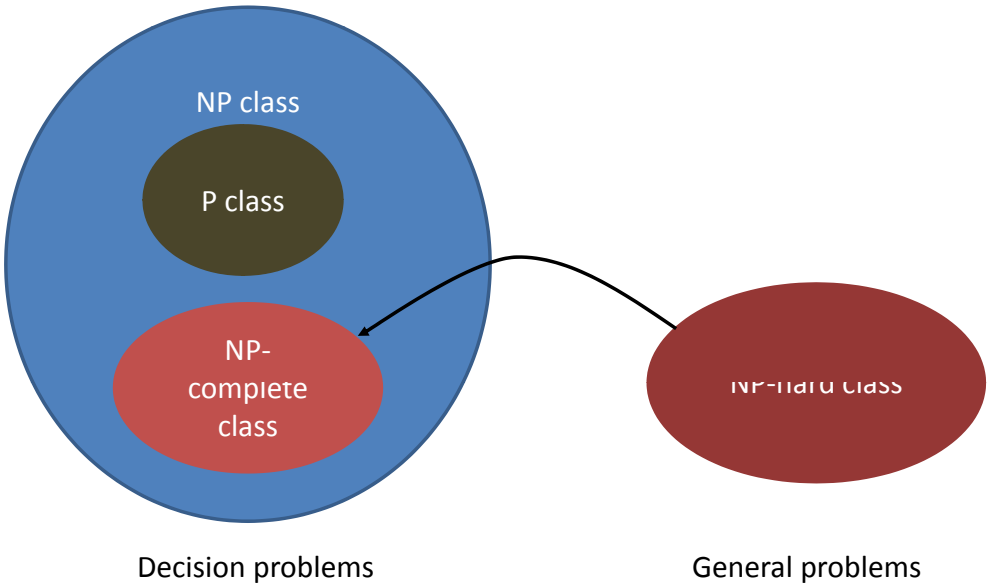
Easy problems are grouped into P class problems, where its worst complexity is bounded by a polynomial function $f(n)$, representing n the input size of the problem instance. Hard problems are grouped into NP class, containing those problems which can be solved in polynomial time by a non-deterministic algorithm. Its used primitives are the following:

- **choice:** proposes a solution
- **check:** verifies (in polynomial time) a solution proposal $\rightarrow$ leads to a *success* or *fail* situation, according to the verification result

The definition of a problem as NP-hard relies on two concepts: the polynomial reduction and the NP-complete class. A given decision problem A is polynomially reduced to a decision problem B if equation 2.1 is satisfied:

$$\forall I_A \in A \exists_{pt} I_B \Rightarrow success_{I_A} \Leftrightarrow success_{I_B} \tag{2.1}$$





where $\exists_{pt}$ implies that the instance I_B can be built in polynomial time with respect of the instance I_A.

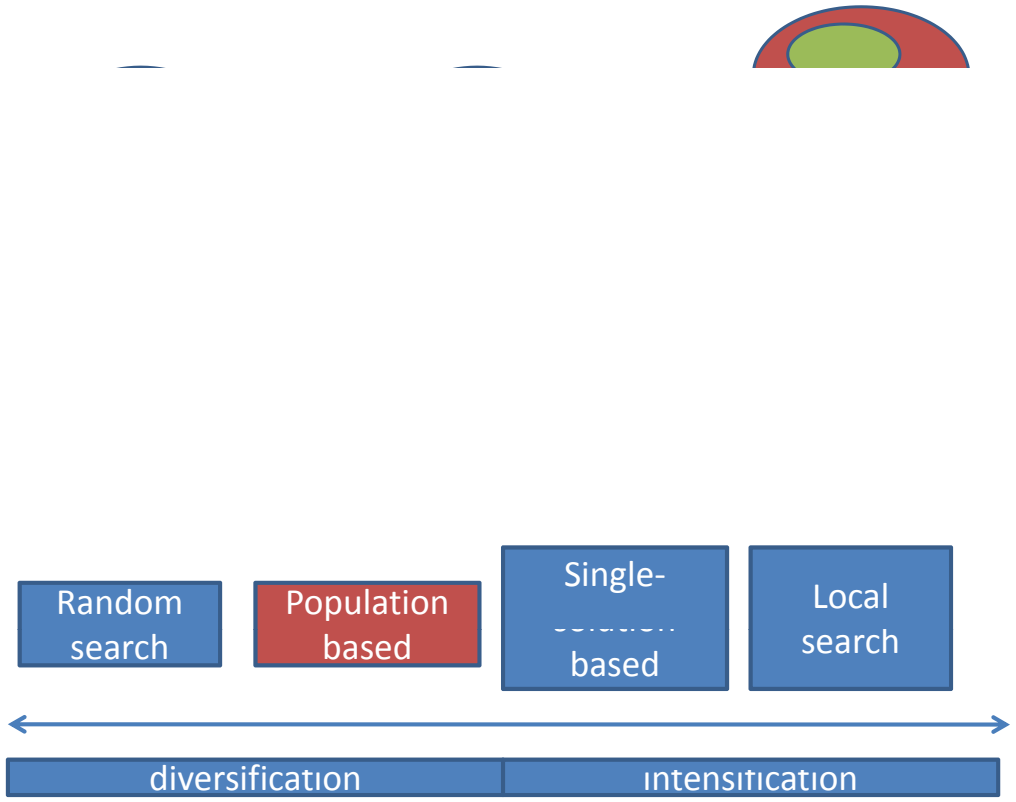
A decision problem $A \in NP$ is *NP-complete* if all other problems of class NP are reduced polynomially to the problem A. Finally, a problem C is *NP-hard* if its associated decision problem is *NP-complete*. Figure 2.3 shows a simple overview over these classifications

There are many academic problem categories which are, inherently, NP-hard, such as sequencing and scheduling, assignment and location, grouping and so on. Real life applications usually fall into this complexity class (only from the presented academic categories is relatively easy to derive many instances of these applications), determining the importance of these problems (and thus, of those techniques which are able to find solutions for them).

Finally, it is important to highlight that the objective of this section is not to focus on the different classifications, but present which kind of problems metaheuristics will deal with and its importance for both academic and real life applications. Regarding this choice, metaheuristics do not only deal with NP-hard problems necessarily, and NP-hard problems do not have to be always approached with metaheuristics. P problems where the power of the exponential function is high and the instance size is also high may require the use of metaheuristics in order to be able to find a solution, while NP-hard problems with a low instance size or particular structures may be resolved with exact or approximation algorithms.

2.3 Metaheuristics

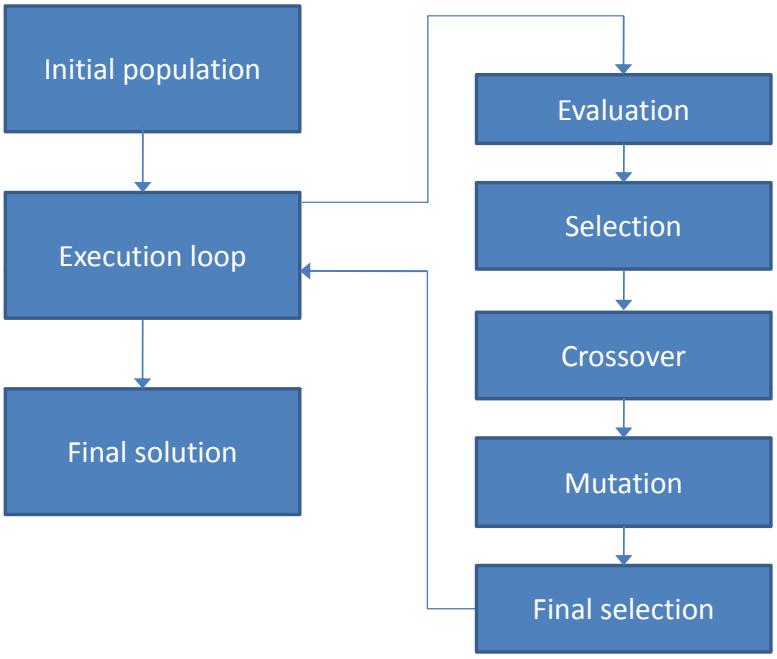
Exact and approximation methods, as introduced in the previous sections, provide with the tools to obtain exact or at least bounded solutions, but such solutions may be unfeasible as the complexity of the problems increases (being particularly true for NP-hard problems). On the other hand, the objective of metaheuristics is to provide satisfactory solutions in a reasonable time.



Any metaheuristic can be basically divided into two different processes: the exploration of the search space and the exploitation of the obtained solutions. These criteria are in conflict and require to be considered jointly. It is interesting to notice a similar definition will be presented, in following sections, for Multi-Objective approaches, first regarding the objective function in section 2.4.3 and afterwards as part of a deeper analysis of these approaches in section 2.5.1. The general idea is to balance the search for better solutions in previously found "good" zones (exploitation, also named intensification) with the search over previously unvisited zones (exploration, also named diversification), in order to guarantee an even exploration over the different zones of the search space. Figure 2.4 shows an example of these processes.

Different techniques provide different coverage rates of these processes. Figure 2.5 provides an overview over the coverage of these processes for different metaheuristic classes.

The random search technique is a pure diversification technique with no use of the quality of previously found solutions. Moving towards exploitation schemes we may use population based techniques, which evolve a whole population of solutions at every step. This approach includes some exploitation due to the evolution introduced, but at the same time is still focused on the diversification over the search space. Single solution based approaches evolve a single solution at every step. This allows them to intensify the search in the local regions and makes them intensification oriented approaches. Finally, the most basic approach to local search is a pure intensification technique where the current solution is replaced at every step with a neighboring one which improves it.



Obviously, the previous classification is a rather rough one. In (Talbi, 2009) the reader may find an exhaustive overview of the different techniques and the evolution which they have gone through over the different years. In this work, we will focus on population based metaheuristics and what is usually called evolutionary computing (Bäck, 1996). It is also interesting to consider that many of the concepts presented in this work can be adapted to different classes of metaheuristics.

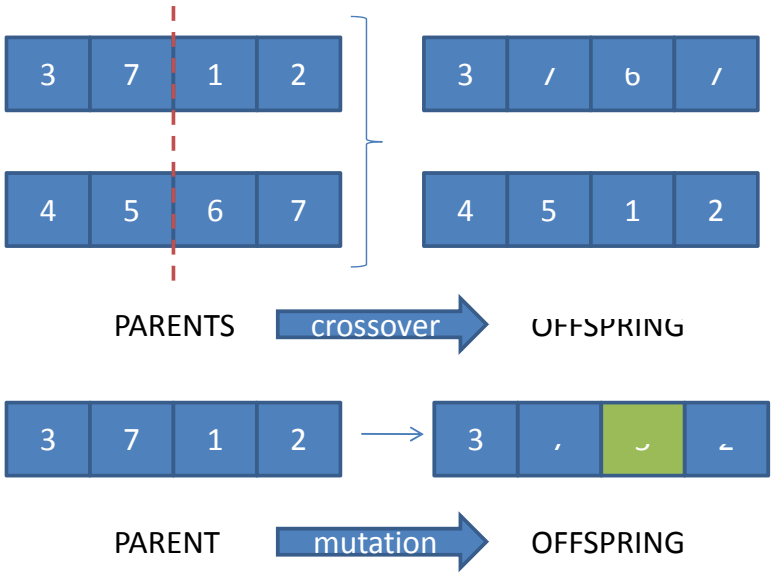
2.4 Evolutionary computing

Population based metaheuristics are defined by the evolution of a whole population of solutions at every step of the algorithm. Some of the techniques included under this metaheuristics' paradigm are scatter search (Glover, 1977), swarm intelligence (Bonabeau et al., 1999) or the one we will focus this work in: evolutionary computing (Bäck, 1996).

The core idea for any evolutionary computing algorithm is to evolve the maintained population by means of crossover and mutation operators, selecting a certain number of individuals from the resulting generation and keeping this process until a certain stopping criterion is met. This process resembles the evolution theories of Darwinism (Darwin, 1859). Figure 6 shows an overview of this process.

The crossover operator takes a number of parent solutions and provides a number of offspring (usually the number of parents and offspring is the same) by recombining certain sections of the parents into their offspring. On the other hand, the mutation operator chooses a single solution and changes a portion of it, providing an offspring. Figure 2.7 shows very simple examples of these processes.

The precursors of evolutionary computation in general are genetic algorithms, a term first



introduced in (Holland, 1975), even though some similar ideas of crossover and mutation as evolution operators were being developed in (Rechenberg & Eigen, 1973) and (Schwefel, 1977). In fact these ideas respectively lead to two of the different approaches usually included under evolutionary computation: genetic algorithms and evolution strategies. Along with the two previous categories, two additional clearly differing approaches can be included: evolutionary programming (Fogel, 1963) and genetic programming (Koza, 1992).

The main original difference between the different approaches was the aim and the solution representation, which lead to different operators

- **Genetic algorithms:** Aimed at discrete optimization, they originally used bit string as encoding for the solutions. The crossover operator is usually based on a number of points chosen from the parents and the mutation one performs bit flipping with a certain probability.
- **Evolution strategies:** Aimed at continuous optimization, they base their representation on real valued vectors. This differed representation lead to crossover operators more scarcely used (being based on discrete approaches) and mutation operators based on Gaussian perturbations of the original values. It is noticeable that evolution strategies allow self-adaptation mechanisms, evolving not only the solution, but also their guidance parameters (typically, their mutation rate).
- **Evolutionary programming:** Aimed at machine learning, their representation is based on finite-state machines. These techniques do not use crossover operators and focus the evolution on the mutation operator, which, in a similar way to evolution strategies, introduces Gaussian perturbations.
- **Genetic programming:** Aimed at machine learning as well, genetic programming uses sparse trees as its representation form. The crossover operators exchange different

subtrees from the parent solutions, and mutation operators introduce random changes in the subtrees.

It is important to highlight that the above comparison is not thorough (for example, the selection mechanisms have not been included) and also that, through the years since their proposal, these techniques have adapted their main attributes according to their uses, being examples of these changes genetic algorithms with different codifications to their original bit string proposal (Dudek, 2006) or evolutionary programming schemes used for continuous optimization (Yang et al., 2006).

According to the previous introduction, the main concepts related to evolutionary programming are the following: representation, initialization, objective function, crossover operators, mutation operators, selection approaches and stopping criteria. We will provide an overview over each of these topics.

2.4.1 Representation

Through the previous sections of this work we have repeatedly used the term solution in a general way, referring both to its encoding and the solution value itself. In evolutionary computation these two meaning are named genotype (solution encoding) and phenotype (solution value). Also, the whole codified solution is frequently referred to as chromosome, whereas each of its units is referred to as gene. The representation chosen has a huge impact on the solution and the algorithm performance, since it usually affects and determines the applied operators.

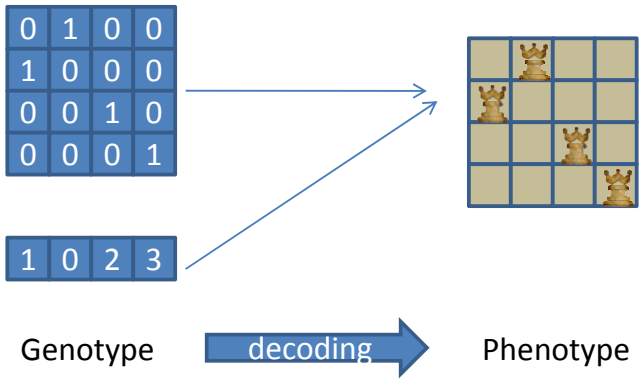
The most usual representation presents a one-to-one relationship between genotype and phenotype values, but this is not necessarily so, since a single phenotype value may be represented by several different genotypes and vice-versa. However, these representations usually impact the performance of the algorithm.

The main representations used by the different approaches of evolutionary computations have already been presented: bit strings for genetic algorithms, float vectors for evolutionary strategies, finite-state machines for evolutionary programming and sparse trees for genetic programming. It is also possible to use mixed representations, a requirement when dealing with global optimizations problems where the variables may have discrete or continuous values.

Most problems can be solved by means of different representations. Thus, choosing the right representation becomes a crucial issue in terms of performance. A very easy example can be found with n queens problem (Letavec & Ruggiero, 2002). This problem, originally stated by Carl Gauss around 1850, tries to determine how to place N queens in an NxN chessboard. The representation issue faced here is how to codify this problem in the most efficient way.

An immediate codification for the genotype would be to use a NxN matrix using 0's for those positions which are empty and 1's for those where a queen is placed. Another possibility is to use the fact that in every column (or row) there must be a queen, and introduce this fact in a permutation codification. Figure 2.8 shows an example of these two different representation approaches codifying the same phenotype.

The second representation provides a much more compact solution representation and, at the same time, introduces some domain knowledge which allows the reduction of the search space: there is one (and only one) queen in every row/column.



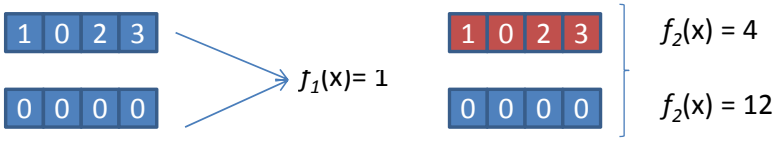
2.4.2 Initialization

The initialization of the initial population may bias the whole process towards success or failure. The default initialization procedure is usually a random initialization, which, depending on the representation used, is performed in different ways. For the default bit string codification of genetic algorithms, the process simply assigns a 0 or 1 value to every gene, whereas for evolutionary strategies those values are assigned randomly according to the upper and lower boundaries of the codified variable. Another common situation is to have a certain alphabet of possible values, according to which the variable (usually codified as an integer) is initialized. For this process, a random number is generated according to the alphabet length. We would like to point out the term random initialization is a terminology abuse, since pseudo-random number generators are used for this task.

The objective of random initialization (regardless of the used representation) is to maximize the initial coverage of the variable space, in order to prevent situations such as early convergence due to local optima or obtaining only a partial inspection of the search space. In the particular case of genetic algorithms, the random initialization is said to achieve the maximal bit-wise diversity. Even so, additional random initialization methods have been proposed, trying to preserve that characteristic, such as (Kallel & Schoenauer, 1997).

The initialization topic has not received as much attention in evolutionary strategies, disregarding its effect as a temporary one for the initial generations (Maaranen et al., 2007). Even so, there have been a number of proposals regarding this topic, such as the quasi-random population initialization presented in (Maaranen et al., 2004), which presented an improved final solution quality, with a limited impact on the convergence speed. This approach's effect was reduced for higher dimensionalities. A thorough comparison of different initialization methods with a focus on quasi-random initialization can be found in (Maaranen et al., 2007). A different alternative may be found in (Rahnamayan et al., 2007), based in the so called opposition-based learning. This concept is related to the consideration of the estimate and opposite estimate at the same time to produce a better approximation for the given candidate solutions. Also, this method tries to overcome some of the difficulties found in quasi-random initialization (namely the difficulty and computational cost of the required processing).

Nonrandom initializations are also possible. This fact was already used in (Grefenstette, 1987). The objective of these approaches is to obtain initial populations which satisfy certain characteristics (usually regarding a good dispersion of their individuals). The main



drawback of nonrandom initializations is their usual high cost (since they have to search for chromosomes which satisfy the imposed restrictions) or the requirement for expertise in the user (in order to avoid that high cost search). These ideas have been implemented in a series of works, such as (Schultz & Grefenstette, 1990).

Examples of different approaches to non random initializations are the simple sequential inhibition process (Diggle, 1983) or the division of the search space into different sub-regions where random initialization is applied independently (McKay et al., 1979). An initial selection from a series of randomly generated individuals was presented by (Bramlette, 1991), and in (Maresky et al., 1995) the relation between non random initialization and re-start strategies is studied and exploited. Finally, an interesting iterative process was presented in (De Garis, 1990), where, according to a set of fitness functions, a population initialization is based on the final population from a previous GA final population evoluted with a different fitness function from the set. This idea is remarkably similar to those presented by $a-priori$ approaches for multi-objective optimization, which will be covered in section 2.5.1. Also, these ideas were applied to constrained optimization by (Schoenauer & Xanthakis, 1993).

2.4.3 Objective function

We may define a single objective optimization problem with equation 2.2

$$f : \chi \rightarrow \Re \; min_{x \in \Re} f(x) \tag{2.2}$$

where $f(x)$ is the objective function. Relating this objective function with the previously introduced concepts, it measures the quality of the phenotype from the genotype value. The definition of this objective function may be a direct reformulation of the problem which is under consideration or a much harder task, in case the objective function has to provide guidance to the solution in an indirect way.

If we go back to the n-queens example presented in the representation section, we might try to define a straightforward objective function which would output a 0 if the solution is valid (the n queens are placed in their proper places) or 1 if there is some incompatibility among them. On the other hand, we might propose a different objective function counting the number of incompatibilities (adding 1 for each incompatibility a queen has with any of the remaining ones). Figure 2.9 shows these two different approaches:

The first objective function obtains the same value for the two different solutions, not being able to provide the search algorithm with any information, while the second approach is able to determine that the first solution is closer to the one the algorithm is looking for.

This example also introduced the concept of restrictions: additional conditions that a certain solution must meet in order to be valid. This leads us to a more general definition of equation 2.2, presented in equation 2.3

$$f : \chi \rightarrow \Re \; min_{x \in \Re} f(x) \text{ such that } \begin{cases} g_i(x) \leq 0 & i = [1 \ldots n] \\ h_j(x) = 0 & j = [1 \ldots m] \end{cases} \tag{2.3}$$

The inclusion of restrictions also introduced additional difficulties regarding how to handle them (Michalewicz, 1995), (Oliver, 2010). The first clear alternative is to penalize the function value according to the violation performed (Homaifar et al., 1994). This alternative can lead to the complete rejection of solutions containing any violation (usually named death penalty (Schwefel, 1993)) or a numerical penalization in the value of the objective function (Joines & Houck, 1994). Another alternative is the use of repairing algorithms (Belur, 1997), (Coello, 2002), which either repair and overwrite the infeasible solution itself (the genotype value is overwritten to a new one which produces a feasible solution, which receives the name of Lamarckian approach) or only the fitness from the repaired solution overwrites the fitness from the unfeasible one (Baldwinian approach). The obvious handicap of repairing algorithms is that establishing them can be a problem of a similar difficulty to the general one which is being solved by the evolutionary algorithm.

There is one more generalization to be performed over the equation stated in 2.3. Up to this point, only one fitness function has been used, but, in general, problems may require the optimization technique to optimize more than one of such functions (potentially in conflict among each other) jointly. Therefore, the general equation for a multi-objective optimization problem (Coello & Lamont, 2004) can be defined with equation 2.4

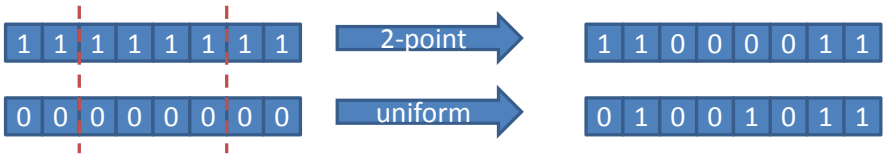
$$\begin{aligned} f_p : \chi \rightarrow \Re, \;\; F(x) = (f_1(x), \ldots, f_k(x)) \; min_{x \in \Re} F(x) \\ \text{such that } \begin{cases} g_i(x) \leq 0 & i = [1 \ldots n] \\ h_j(x) = 0 & j = [1 \ldots m] \end{cases} \end{aligned} \tag{2.4}$$

a clear and complete textual definition can be found in (Osyczka, 1985): a multiobjective optimization problem can be defined as the problem of finding "a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the decision maker."

2.4.4 Crossover operators

Crossover or recombination operators are responsible of integrating the characteristics of two (or, in general, n) parents into their offspring. The main characteristics to consider regarding crossover operators are their heritability, (since their offspring inherit genetic material from all the different parents) which can be strong if identical parents always produce identical offspring, and the validity of the generated offspring (which may not be assured if the problem is constrained, leading to the possible constraint handling techniques discussed in the objective function section of this work).

For linear representations (the ones presented traditionally in either genetic algorithms or evolutionary strategies) not including permutations (representations where a vector of size n includes n different numbers, such as the *n-queens* problem) the traditional crossover



operators are n-point crossover and uniform crossover. n-point crossover chooses randomly the position of n genes in the chromosome and then parents provide their genetic material according to those positions. On the other hand, uniform crossover chooses, for each gene position one of the parent solutions, which randomly provides its genetic material. Figure 2.10 provides an example over these crossovers

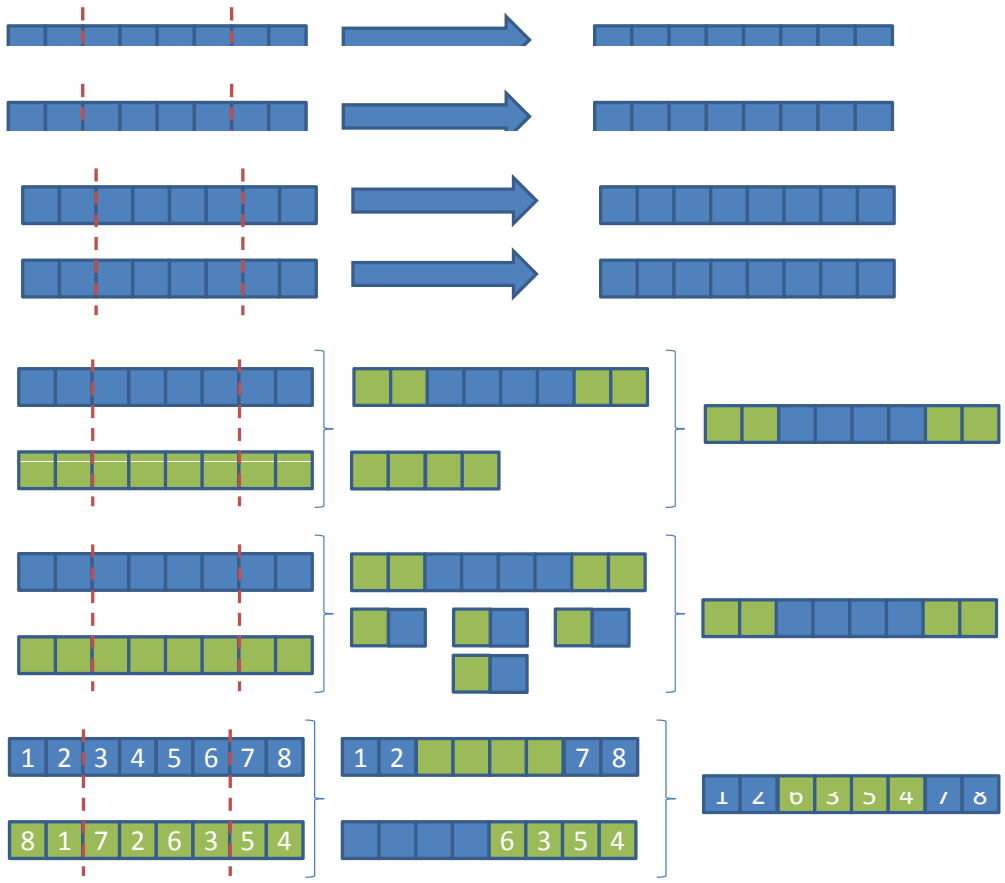
For real valued linear representations (evolutionary strategies) additional crossover operators are defined, trying not to inherit the concrete values from the parents, but numerical values obtained from their combination, according to different procedures. The main approaches to this task are mean-centric operators and parent-centric operators.

Mean centric operators compute the mean of two or, in general, n parents and obtain the offspring value according to different procedures. Intermediate and geometrical crossovers [Mickaleicz et al. 1996] combine the parents' values using either sums or multiplications of their gene values. Unimodal normal distribution crossover (UNDX) (Ono et al., 2003) uses the mean value from $\mu - 1$ parents and also a direction according to the directions of those parents in order to generate the offspring. Finally, Simplex crossover (SPX) (Tsutsui et al., 1999) generates the offspring around the mean in a delimited zone according to the parent simplex.

Parent centric operators obtain offspring closer to the given parents. The main approaches to these operators are simulated binary crossover (SBX) (Deb & Agrawal, 1995) and parent-centric crossover (PCX) (Deb et al., 2002a). SBX performs a weighed sum of the parents values according to a factor derived from an input variable η (where higher values imply the generation of offspring nearer to their parents) and a proposed distribution function. A random number is calculated and then the weighing factor is calculated so that the area under the probability curve of the proposed distribution function is equal to the value of that random number. PCX chooses a parent randomly, obtains its direction vector and the final offspring value is computed according to the value from the chosen parent, its direction vector and the perpendicular distances to the line given by that direction vector of the remaining $\mu - 1$ parents.

The techniques presented for linear representations would provide, applied to permutation representations, unfeasible solutions. That involves the requirement for problems choosing these representations of particular crossover operators. We will cover here three alternatives to these operators: the order crossover (OX) (Davis, 1985), the partially mapped crossover (PMX) (Goldberg & Lingle Jr, 1985) and the two-point crossover (Wiese & Glen, 2003). The literature shows additional strategies (and even those presented can be performed in different ways) but we only want to give an illustrative set of the mechanisms introduced to deal with permutation representation.

The order crossover (OX) chooses two random positions in the chromosomes, and fills the offspring with the genes from the first parent at the same positions. Afterwards, the rest of the gene positions are filled with the elements of the second parent not already included



starting from the second selected position. Figure 2.11 shows this process. The partially mapped crossover (PMX) starts again with the selection of two random positions. The genes from the first parent between those positions are copied into the offspring, and a mapping is obtained from the elements of the second parent between those positions and them. The elements of the second parent not between the given positions are copied into the offspring, and the mapped elements are replaced with their respective ones. This process is explained in figure 2.12. Finally, two-point crossover selects two random positions, copies from the first parent the elements outside those positions at their same positions and fills the positions between the two selected ones with the elements not already included of the second parent (starting from its beginning gene). An example is shown in figure 2.13.

Special crossover operators have been also designed for the genetic programming approach and its sparse tree representations (evolutionary programming does not use crossover operators, only mutation ones). Since the application problems of this work will not be centered in this representation, we will not cover here these operators. However, for an overview of them, the reader may consult (Spears & Anand, 1991).

2.4.5 Mutation operators

Mutation operators are unary operators which introduce small changes in selected individuals from a given population (these operators are often applied after the crossover ones over the resulting offspring). The underlying idea is that a small change in the genotype should also produce a small change in the phenotype (locality). The design of a mutation operator must also consider the properties of ergodicity (every solution of the search space must be reachable) and validity (the resulting solution after the operator application must be valid).

The validity property can be an issue in constrained domains, and different strategies may be defined to deal with it, as shown in section 2.4.3.

mutation_rate is the common parameter for the different operators and representations, being the probability of mutating a variable. Its values have been set according to different experimental results. In (Bäck, 1993, 1996; Mühlenbein & Schlierkamp-Voosen, 1993) the proposed value is $1/n$, where n is the number of variables (genes) in the given solutions, showing its good results for different representative sets of problems. This *mutation_rate* implies that, in average, one gene is mutated each time that the mutation operator is performed over a given individual.

This *mutation_rate* is the basis for the application of mutation operators to binary, discrete and permutation representations. In binary representations, every gene has only two possible values, leading to a mutation which only changes the variable value to its complementary value. However, it was very common to use binary representations to codify real or integer values, linking the mutation operator to the codification used (Chakraborty & Janikow, 2003). This has led to the recommendation of real-valued vectors instead of binary representations and their associated mutation operators (Michalewicz, 1996). Discrete representations involve a change in the gene value with a different one in the associated alphabet (the distances between the different alphabet members may be different, leading to the inclusion of a *mutation_step* parameter which we will define later in this section) and, finally, permutation representations apply mutation operator exchanging the position of two different gene values.

Mutation operators applied to real-valued representations introduce two new considerations: the *mutation_step* parameter and the static or dynamic nature over the parameter values. The *mutation_step* parameter represents the size of the introduced mutation, being a hard value to set. In general, small values are considered to obtain good results in the long run, while bigger values may produce, if successful, much quicker results. This has led to the proposition of operators combining a biased use of these two approaches, producing smaller steps with a greater probability (Muhlenbein, 1994). The definition of these mutation steps can be performed by different processes, most of which are based on the sum of a certain value to the original one from the gene, being this value defined by processes such as random, Gaussian or polynomial operators.

A uniform random mutation operator obtains a value in a certain range, which is defined by the user. A Gaussian mutation operator uses values generated by means of normal distributions, $N(0, \sigma)$, to mutate the different genes values. This is the most extended approach in evolutionary strategies (and probably in evolutionary computing in general), and the definition of the *mutation_step* is performed in a direct way by means of the σ parameter. Finally, the polynomial mutation operator (Deb & Goyal, 1996) adds a certain value to the original value according the polynomial distribution, which introduces a η_m index.

One of the most interesting particular approaches of real-coded representations is the possibility of evolving the mutation parameters along with the solutions. The first approaches to this task were proposed early in the development of evolutionary strategies (Schwefel, 1981), and have been evolving ever since. The general idea behind this task is to obtain an adaptation of the size of the mutation step and also of the direction of that mutation. This has led to the development of techniques achieving the adaptation of n step sizes (Ostermeier et al., 1994) or n step-sizes with one or n directions (Hansen et al., 1995).One of the most successful recent approaches is the Covariance Matrix Adaptation Evolutionary Strategy

(CMA-ES) (Hansen & Kern, 2004; Hansen & Ostermeier, 1996). This strategy adapts the covariance matrix of the multivariate normal mutation distribution, introducing the leading ideas of giving a higher priority to the most successful step-sizes and also memorizing the time distribution path of the mean, in order to introduce additional control over the step-size.

2.4.6 Selection and replacement strategies

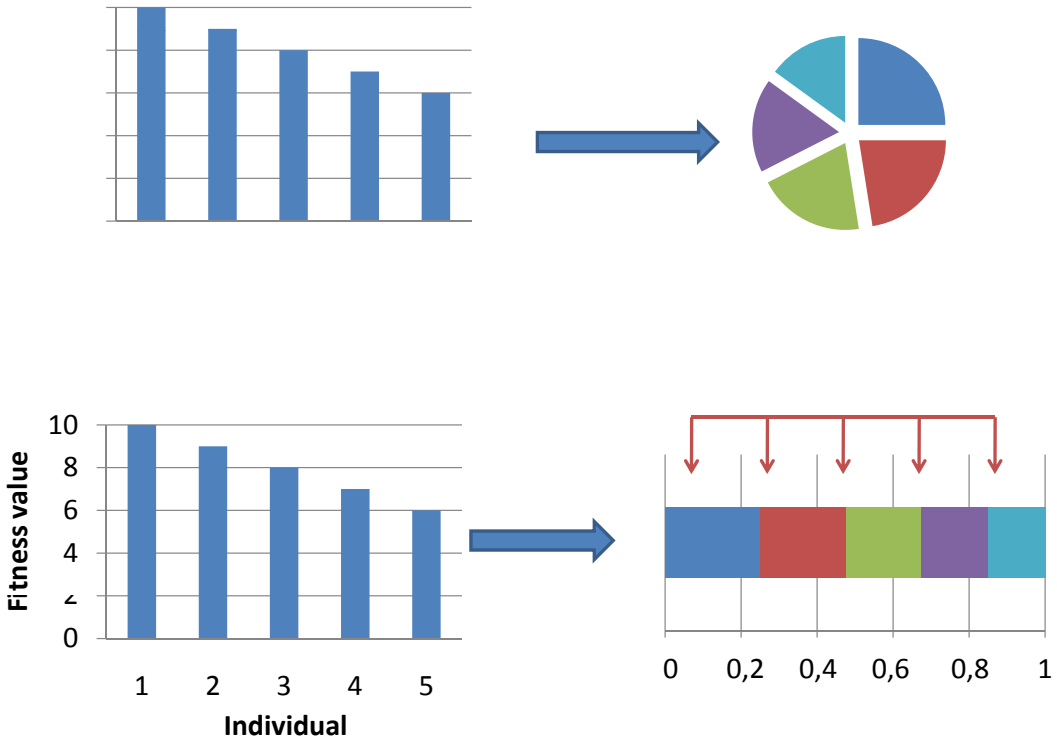
Selection and replacement strategies are closely linked (in fact, in figure 2.6 we referred to the replacement strategy as final selection) and rely on the concept of elitism (De Jong, 1975), even though in different ways (at least according to their application). In general, elitism implies preserving the advantages of the best individuals. These advantages, related to selection strategies, are applied according to their reproduction chances, and, related to replacement strategies, according to their survival chance.

Selection strategies, thus, must provide better chances of becoming parents to these individuals with a better fitness value, but also giving a chance to those worse adapted individuals which may still introduce valuable genetic material into the algorithm (Bäck et al., 2000). This is linked with the idea of diversity preservation, and even more, with the initial discussion presented in section 2.3 about the exploration /exploitation focus. In general, selection strategies do not only determine which individuals are selected for mating, but may also determine how many offspring they will produce (the number of offspring is also related to the replacement strategy used).

The fitness assignment is the prior process to any selection strategy, and may be performed in an absolute or relative way. Absolute assignment applies the fitness function value directly to the individual, while relative (or rank-based) assignment (Whitley et al., 1989) determines a sorting over the population assigning ranks to the different individuals and applies the selection techniques over these values. Rank-based methods perform a transformation over the fitness value according to linear or non-linear functions (Pohlheim, 1995). They introduce a parameter to control explicitly the selective pressure over the population, and have been reported to be more robust the fitness proportional assignment (Bäck & Hoffmeister, 1991). The most extended selection techniques (which are independent to the fitness assignment technique) are roulette wheel selection, stochastic universal sampling (Baker, 1987) and tournament selection(Blickle & Thiele, 1995; Goldberg & Deb, 1991).

Roulette wheel selection is a stochastic selection algorithm based on the assignment of a probability to each individual based on its fitness value, and projected in a contiguous segment. Once that has been performed, a uniform random number in the [0, 1] range is generated, and the appropriate individual, according to its value in the segment, is selected (figure 2.14). This process is repeated until the whole mating population has been selected. One of the handicaps of this approach is the bias introduced by outstanding individuals, which may lead to premature convergence of the algorithm.

Stochastic Universal Sampling tries to correct that introduced bias in the roulette wheel selection mechanism. In order to do this, it introduces a series of equally spaced pointers, with a distance between each pair of them of $1/n$, where n is the number of individuals to be selected. When the random number is generated, the first pointer is moved to that position, and the remaining ones properly located according to their distance. Each of the pointers selects an individual, so that with a single random number the whole mating population is selected. This process is shown in figure 2.15



Tournament selection requires the choice of a tournament size k, determining how many individuals will participate in each selection. To obtain each individual in the selected set, k individuals are randomly chosen and the best one among them is selected for the mating pool (figure 2.16). This process is repeated until all the different required individuals have been selected.

Replacement strategies perform the required selection from the offspring of the current generation, i, to the initial population of the following generation, $i+1$. Elitist replacement strategies may be used, along with the different selection techniques approached in this section. However, elitism may lead to premature convergence situations, and thus, different replacement strategies have been proposed. The canonical genetic algorithms proposed in (Holland, 1975) performed a generation replacement, where an offspring containing a number of individuals equal to the original population was produced and replaced the original

one. However, this led to considerations in (De Jong, 1975) about the possibly wasted good solutions and with this approach. This leads to strategies performing only a partial replacement of the original population (what is usually called the generational gap), up to incremental (or steady-state) replacement strategies, where only one chromosome is replaced in each generation (Davis & Mitchell, 1991). The effect of replacement techniques in steady-state approaches and its relation with the diversity maintenance is analyzed in (Lozano et al., 2008).

Evolutionary strategies typically perform two approaches to the replacement issue (Reeves, 2003). The first one, typically referred to as (λ, μ), implies the generation of μ offspring, where μ is greater than the number of parents, λ. Thus, after the crossover and mutation procedures have been applied, the λ best individuals from the offspring are selected and turn into the starting population of the following generation. The second replacement strategy, usually named $(\lambda + \mu)$, implies that both populations, parents and offspring, are merged, and, afterwards, the best λ individuals are chosen from the overall population. This second approach introduces, in general, a stronger elitism into the resulting algorithm.

2.4.7 Stopping criteria

To introduce the lack of attention dedicated to convergence and stopping criteria, in general, we would like to cite the paragraph included in (Chipperfield et al., 1994): *"Because the GA is a stochastic search method, it is difficult to formally specify convergence criteria As the fitness of a population may remain static for a number of generations before a superior individual is found, the application of conventional termination criteria becomes problematic A common practice is to terminate the GA after a pr-specified number of generations and then test the quality of the best members of the population against the problem definition If no acceptable solutions are found, the GA may be restarted or a fresh search initiated"*

Some of the most extended characteristics of stopping criteria are included in the previous cite: difficulties in establishing stopping criteria, which leads to criteria which establish a-priori boundaries (such as the number of generations or function evaluations) or dynamic ones based on previous knowledge of the problem (which allows the determination of whether a solution is acceptable or not). In (Talbi, 2009) these approaches are classified as *static* (use of a-priori boundaries) or *dynamic* (use of problem knowledge to determine in run-time when the found solution is acceptable). It also points out to the use of population characteristics to determine the stopping situation, particularly the population diversity, but without any concrete technique or citation. In (Reeves, 2003) the same principles are explained, including more detail about the possible approaches to diversity and stagnation concepts (which can be applied to genotype, phenotype or fitness values). In (Coello, 2000) the most extended stopping criterion was stated to be an a-priori chosen maximum generation, and brand new algorithms, including state of the art developments in all their individual aspects, such Hype (Bader & Zitzler, 2011) still maintain that default choice.

Practical approaches of these general development strategies can be found in works such as (Zielinski et al., 2005). In this work, a classification and division among stopping criteria is established, particularly for differential evolution (DE) (Fleetwood, 1999) and Particle Swarm Optimization (PSO) (Kennedy, 2006). The established classes are the following:

1. *Reference criteria*: These criteria require the *a-priori* knowledge of the optimum solution, and thus they may be inapplicable to new real problems. The algorithm is stopped

after a certain percentage of the population has converged to the given optimum (Espinoza et al., 2001).

2. *Exhaustion-based criteria*: The algorithm is stopped after a certain generation, number of function evaluations or CPU time is reached.

3. *Improvement-based criteria*: The algorithm is stopped if only small (or none at all) improvements are made over a certain time. These criteria can be based on several measures: the improvement of the best objective function value (Van Den Bergh, 2006), the improvement of the average objective function value (Espinoza, 2003), no acceptance or improvement in the neighborhood (these last criteria are less generalizable to different families of algorithms).

4. *Movement-based criteria*: The algorithm is stopped if the movements in the population fall below a certain pre-established threshold. Again these movements may be measured according to two different perspectives: movements with respect to the average function value (or, in other terms, movements in the objective space) or movements with respect to the positions (or, in other terms, movements in the parameter or variable space).

5. *Distribution-based criteria*: The algorithm is stopped after a certain distance measure falls below a certain threshold. The basis of these criteria is the idea that all individuals tend to converge to the optimum, so their closeness indicates the termination of the algorithm. These distance measures can be based on different concepts: the distance of every vector (individual) of the population to the best vector, the distance of a certain percentage of the individuals in the population, standard deviation of the vectors in the population (Zaharie & Petcu, 2005) or the distance between the best and worst individual in the population (Babu & Angira, 2003).

6. *Combined criteria*: Different function features may require different stopping rules to guarantee a satisfactory criterion. A most common combined criteria includes some exhaustion-based criteria along with some of the other different alternatives available, to guarantee a certain *a-priori* computational threshold in case the complementary criterion fails to be triggered. An example of these combined criteria is OCD (Trautmann et al., 2009), which will be analyzed in detail in section 2.7.1.

2.5 Multi-Objective Evolutionary Algorithms

2.5.1 General concepts

The concept of multiobjective optimization was introduced in section 2.4.3. Basically, it is an optimization problem, which, instead of a single function to be maximized or minimized, requires a whole set of n functions to be optimized jointly (equation 2.4. Currently, the most extended approach of evolutionary computation to multi-objective optimization (Coello et al., 2007; Deb, 2001) relies on the Pareto Optimality Theory (Ehrgott, 2005), even though different approaches have been previously explored and exploited, such as aggregating functions (Surry et al., 1995) or the inclusion of the decision maker's choices within the evolutionary cycle (Fonseca et al., 1993).

Pareto-centered approaches are based on the original ideas from the works of (Edgeworth, 1881) and (Pareto, 1896), relying on the Pareto optimality concept (definition 2.5.1).

Definition 2.5.1 (Pareto Optimality): *A solution* $\vec{x} \in \Omega$ *with* $\vec{u} = F(\vec{x}) = (f_1(\vec{x}), \ldots, f_n(\vec{x}))$ *is said to be Pareto Optimal with respect to to* Ω *if and only if there is no* $y \in \Omega$ *with* $\vec{v} = F(\vec{y}) = (f_1(\vec{y}), \ldots, f_n(\vec{y}))$ *such that* $\vec{v}$ *dominates* $\vec{u}$

This definition relies on the Pareto dominance concept, presented in definition 2.5.2

Definition 2.5.2 (Pareto Dominance): *A vector* $\vec{u} = (u_1, \ldots, u_n)$ *is said to dominate a vector* $\vec{v} = (v_1, \ldots, v_n)$ *if and only if* $\vec{u}$ *is partially less than* $\vec{v}$ *This relation is expressed as* $\vec{v} \preceq \vec{u}$ *and defined in equation 2 5*

$$\vec{v} \preceq \vec{u} \leftrightarrow \forall i \in \{1, \ldots, n\} u_i <= v_i \wedge \exists i \in \{1, \ldots, n\} : u_i < v_i \tag{2.5}$$

Definitions 2.5.1 and 2.5.2 allow the relationship between the Pareto optimality concept with the genotype and phenotype concepts used in evolutionary computing (presented in section 2.4.1), or Pareto Optimal Set (for the genotype representation) and Pareto Optimal Front (for the phenotype's associated values)

Definition 2.5.3 (Pareto Optimal Set): *for a given Multi-Objective Problem with the associated set objective functions* $F(X) = (f_1(x), \ldots, f_n(x))$ *the Pareto Optimal Set, represented as* P^**, is defined in equation 2 6*

$$P^* := \{\vec{x} \in \Omega \mid \neg\exists \vec{y} \in \Omega F(\vec{y}) \preceq F(\vec{x})\} \tag{2.6}$$

Definition 2.5.4 (Pareto Optimal Front): *for a given Multi-Objective Problem with the associated set objective functions* $F(X) = (f_1(x), \ldots, f_n(x))$ *and the correspondent Pareto Optimal Set* P^**, the Pareto Optimal Front, represented as* PF^**, is defined in equation 2 7*

$$PF^* := \{\vec{u} = F(\vec{x}) \mid \vec{x} \in P^*\} \tag{2.7}$$

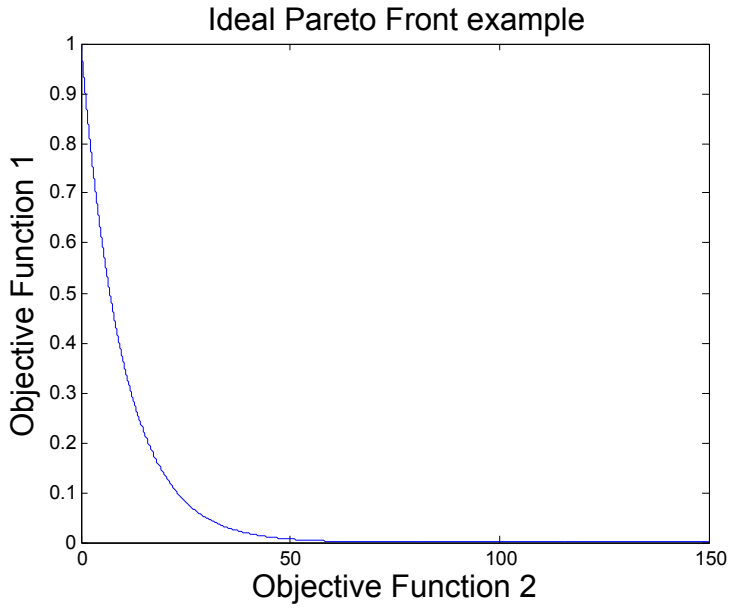
The Pareto Optimality can be weak or strict, according, respectively, to definitions 2.5.5 and 2.5.6

Definition 2.5.5 (Weak Pareto Optimality): *A point* $y \in \Omega$ *is a weakly Pareto Optimal if there is no* $x \in \Omega$ *such that* $f_i(x) < f_i(y)$*, for* $i \in \{1, \ldots, n\}$

Definition 2.5.6 (Strict Pareto Optimality): *A point* $y \in \Omega$ *is a weakly Pareto Optimal if there is no* $x \in \Omega, x \neq y$ *such that* $f_i(x) \leq f_i(y)$*, for* $i \in \{1, \ldots, n\}$

In the context of a Multi-Objective Evolutionary Algorithm (MOEA) progress, additional definitions are required, as presented in (Van Veldhuizen, 1999). These definitions include the current Pareto set (formalized as $P_{current}(t)$ referred to a given generation t). Additional interesting definitions may be related to the maintenance of secondary populations (or archives) which keep the known Pareto solutions (an important example regarding this approach is the SPEA2 algorithm (Zitzler et al., 2001)), defined as P_{known}, or the true Pareto

Figure 2.17: Ideal Pareto Front Example



solution in the computational domain (usually a subset of P^*), defined as P_{true}. Obviously, according to the initial definitions, all these solution sets have their associated Pareto Fronts.

Presented¡ definitions dealt with the optimality concepts from a mathematical point of view, not taking into account the acceptable compromise solutions concept and the decision maker (DM) impact on the overall process (Fonseca & Fleming, 1997). To illustrate these factors, figure 2.17 shows an ideal Pareto Front behavior. The solutions presented in the figure are optimal according to definition 2.5.1, which means that improving one objective function degrades the value of the other one. However, many of these solutions may not be valid according to the decision maker's criteria. This leads to the fact that there are usually many preferences not codified, requiring, as the final solution obtained, to perform a sampling of the final obtained Pareto Front. The evolutionary process ends with the definition of the final set of solutions according to its Pareto Front values, $\vec{u} \in PF^*_{known} \subseteq PF_{known}$. A real example regarding the previous concepts is the two-bars symmetric plane truss (Rao, 1987).

The approach to finding the solutions for a MOP can be based, basically, on three different approaches: optimizing the most important objective function, obtain an aggregated function according to the importance of the different objective functions or obtain the complete Pareto Front by means of MOEAs. These approaches are the basis for the classification of the different MOEA approaches, which includes the following classes:

- **A priori techniques:** These techniques require the DM, in general, to define the importance of the different objective functions in the MOP. The MOP is, with the use of these importance factors, reduced to a single objective optimization problem. It is important to realize that the determination of these relative importance factors (usually in the form of weighing values) can be a particularly hard issue to deal with, and that the overall quality of the final obtained solution relies heavily on this decision. Examples of this approach are lexicographic ordering (Fourman, 1985) and aggregating functions, which can be linear (Surry et al., 1995) or non-linear (Horn et al., 1993)

- **Progressive techniques:** These techniques require the direct interaction of the DM during the EA search process, combining cycles of search and decision making. As explained in (Coello et al., 2007), pp 70-71, there are surprisingly reduced number of examples of these techniques examples available in the literature, probably due to the additional time required by researchers and DMs. As shown in (Van Veldhuizen & Lamont, 1998) many approaches are based on Fonseca's Multi-Objective Generic Algorithm (MOGA) (Fonseca et al., 1993), such as (Sette et al., 1997).

- **A posteriori techniques:** *A posteriori* techniques seek for P_{true} and PF_{true} (Horn, 1997), trying to perform a search as widespread as possible to generate the largest possible number of elements from the Pareto Set. One of the advantages of this approach is that several DMs with different criteria may choose different solutions without any additional search computation. The main techniques dealing with this approach are independent sampling (Fourman, 1985), criterion selection (Schaffer, 1985), aggregation selection (Ishibuchi & Murata, 1998) and Pareto sampling (Goldberg et al., 1989)

2.5.2 Goals and design features

According to the presentation of the MOEA field in section 2.5.1, the four main goals of a MOEA, as stated in (Coello et al., 2007), are the following:

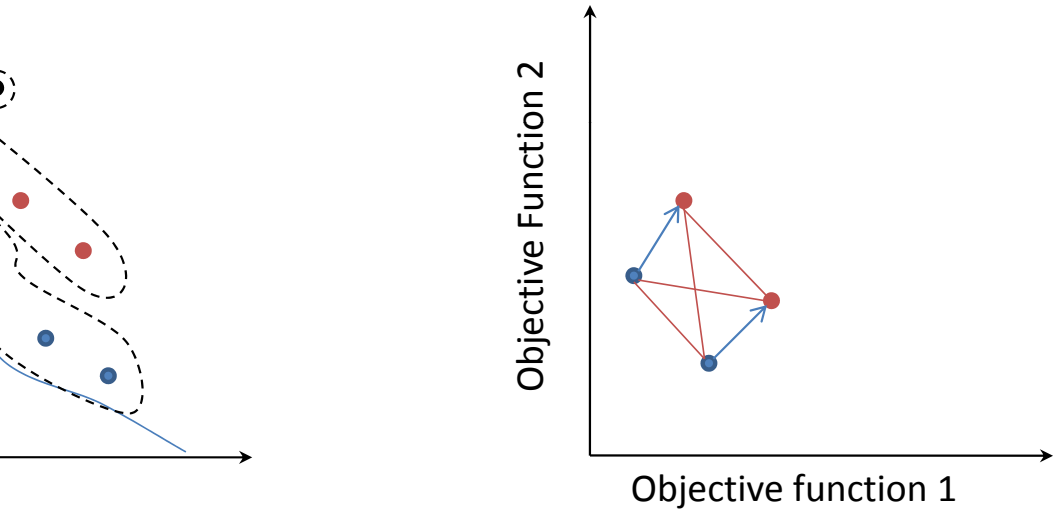
- Preserve non-dominated points
- Guide the progress of PF_{known} towards PF_{true}
- Maintain the diversity of the obtained solutions (at genotype level with P_{known} and / or phenotype level with PF_{known})
- Provide the DM with a limited number of PF_{known} points

In fact, these four goals can be easily related to the concepts already introduced in the general presentation of evolutionary computation. The first two ones deal directly with the concept of elitism and how the algorithm evolves to the best solution (in this case represented by PF^*). The third one is the required diversity preservation (which in single objective optimization was presented as the way to prevent early convergence and discussed according to its relationship with elitism and selection mechanisms in section 2.4.6), while the fourth is particular to the Multi-objective domain, since single-objective algorithms provide a single solution as their output. This section will deal with the tools designed to achieve these objectives.

Dominance Based Ranking

The dominance concept was presented in definition 2.5.2. An important quality of this relationship is that it is not a partial order, but a strict partial order. This implies that, when comparing two solutions $\vec{a}$ and $\vec{b}$, one may dominate the other or they may be incompatible (none of the two dominates the other). Figure 2.18 shows an example where the blue arrows indicate dominance of one solution over the other and the red lines indicate incomparable solutions.

Figure 2.18: Dominance examples

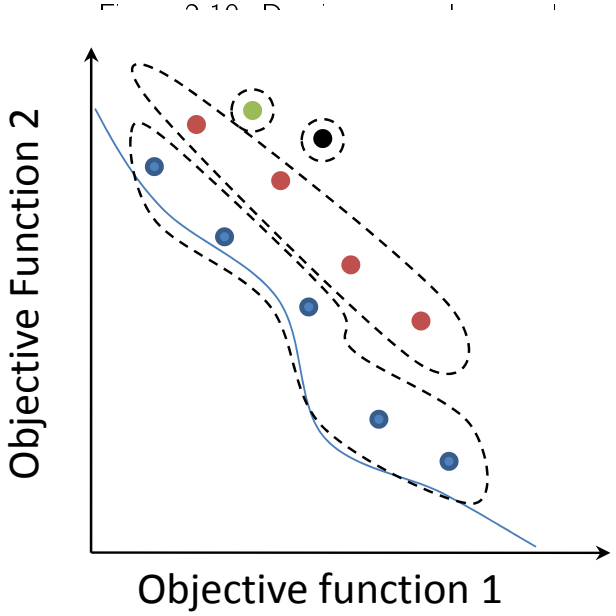


The selection or the Pareto optimal set requires the application of an ordering technique. Several ranking methods have been proposed to deal with this task. These ranking methods provide a rank to a solution according to its objective functions values, according to the following procedures:

- **Dominance rank:** The rank value is assigned according to the number of individuals which the analyzed solution is dominated by. A value of one is added to that number. An example is shown in figure 2.19, where the values with the same rank are presented with the same color and grouped with a dotted line. A relevant algorithm which makes use of it is the Niched Pareto Genetic Algorithm (NPGA) (Horn et al., 1994).
- **Dominance count:** The rank value is assigned according to the number of individuals which the analyzed solution dominates. An example is shown in figure 2.20, following the same presentation criterion as dominance rank. A relevant algorithm which makes use of it is the Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas & Deb, 1994).
- **Dominance depth:** The rank value is assigned according to non-dominated front in which the solution is located. To calculate this value, the first front of non-dominated solutions is found, assigned rank 1 and removed from the population, following this procedure until all the solutions in the population have been ranked. An example is shown in figure 2.21, with the same presentation criterion as the previous ranking techniques. A relevant algorithm which makes use of it is the Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler & Thiele, 1999).

Diversity Preservation

The objective of the dominance ranking techniques presented is to guide P_{known} towards P_{true}. The additional capital goal, obtaining a distribution over the Pareto front as uniform as possible, is faced by diversity preservation techniques. The general idea of these techniques



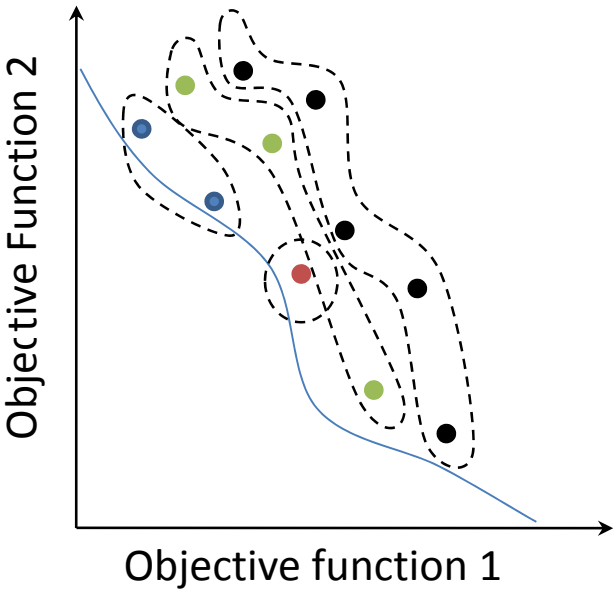
is to "push away", during the evolution of the population, solutions which are very close to each other so that the final front obtained is as spread as possible. This can be performed in different ways:

- **Weight Vector:** This approach uses a vector set in the objective function space in order to bias the search and move away solutions from their neighbors, using for this objective the change in the values of a certain set of weights (Ulungu et al., 1999).

- **Fitness sharing / niching:** This approach is based in the definition of a neighborhood (or niche, which provides the technique with its name) according to a σ_{share} value. Different topologies may be used, such as grids, and also different density estimation criteria, such as kernel (Fonseca et al., 1993), nearest neighbor (Deb et al., 2002b), or histogram approaches (Corne et al., 2000).

- **Crowding / clustering:** This approach is based on an idea similar as the one used by niching: the solutions are selected by means of a crowdedness metric applied to their region. An example is the crowding distance introduced in NSGA-II (Deb et al., 2002b). Since clustering techniques rely on this same idea of grouping solutions minimizing the distance within a cluster and maximizing the distance to additional clusters, these techniques can also be used as part of a diversity preservation scheme (Zitzler et al., 2001)

- **Restricted mating:** This approach, once again, is based on the fitness sharing principles of measuring the density in a determined zone, but, instead of applying it to the final selection procedure of the algorithm, the results are applied to the selection of the mating parents. This way, the parameter σ_{mate} determines a minimum distance that must separate two individuals in order to be able to mate them (Lu & Yen, 2003)



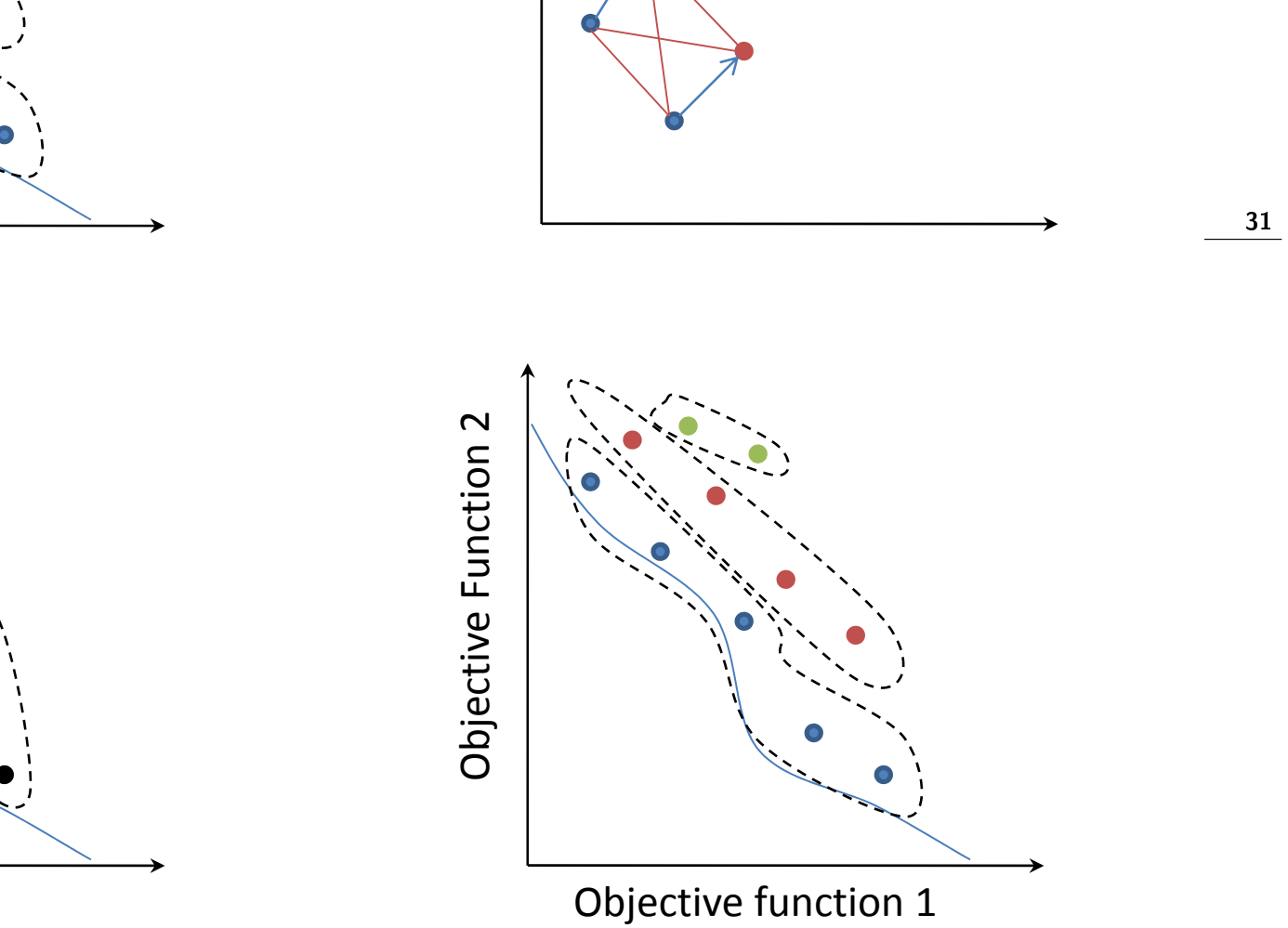
2.6 MOEA Quality Assessment

At the beginning of section 2.1 the initial proposal of exact versus approximate (which include evolutionary algorithms) methods was presented. Approximate methods do not provide with the optimal solution, neither with a known boundary in the approximation of its value. This leads to the need of determining which solution, among different MOEA approaches, is better for a given problem. The term better, in this case, concerns both the quality of the outcome and the amount of resources required to obtain that output. Current section we will deal with this quality issue.

Comparing the quality assessment issue of MOEA with their single-objective versions, the difficult of the Multi-objective approach is clear: a single objective solution quality can be easily determined according to its objective function value, leading to the output of a single solution by the algorithm. However a MOEA outputs a certain number of compromise solutions among its different objective functions, making the assessment of the quality of a given Pareto front much harder. Relating this quality assessment to the goals presented in section 2.5.2, it is possible to consider whether the closeness of the PF_{known} to the PF_{true} is more important or less, for example, than the spread of its solutions. An example regarding this issue is presented in figure 2.22.

2.6.1 Quality indicators

In order to formally deal with this comparison issue, it is necessary to define the different levels of Pareto dominance (Zitzler et al., 2002) (presented in its general form in definition 2.5.2). Definition 2.6.1 presents the dominance relationships between two different solution vectors, whereas definition 2.6.3 extends these concepts to dominance relationships between approximation sets.



Definition 2.6.1 (Dominance relations): *Let Z be the n-dimensional objective space and $\vec{z}^1 = (z_1^1, \ldots, z_n^1), \vec{z}^2 = (z_1^2, \ldots, z_n^2)$, the following dominance relations are defined on Z:*

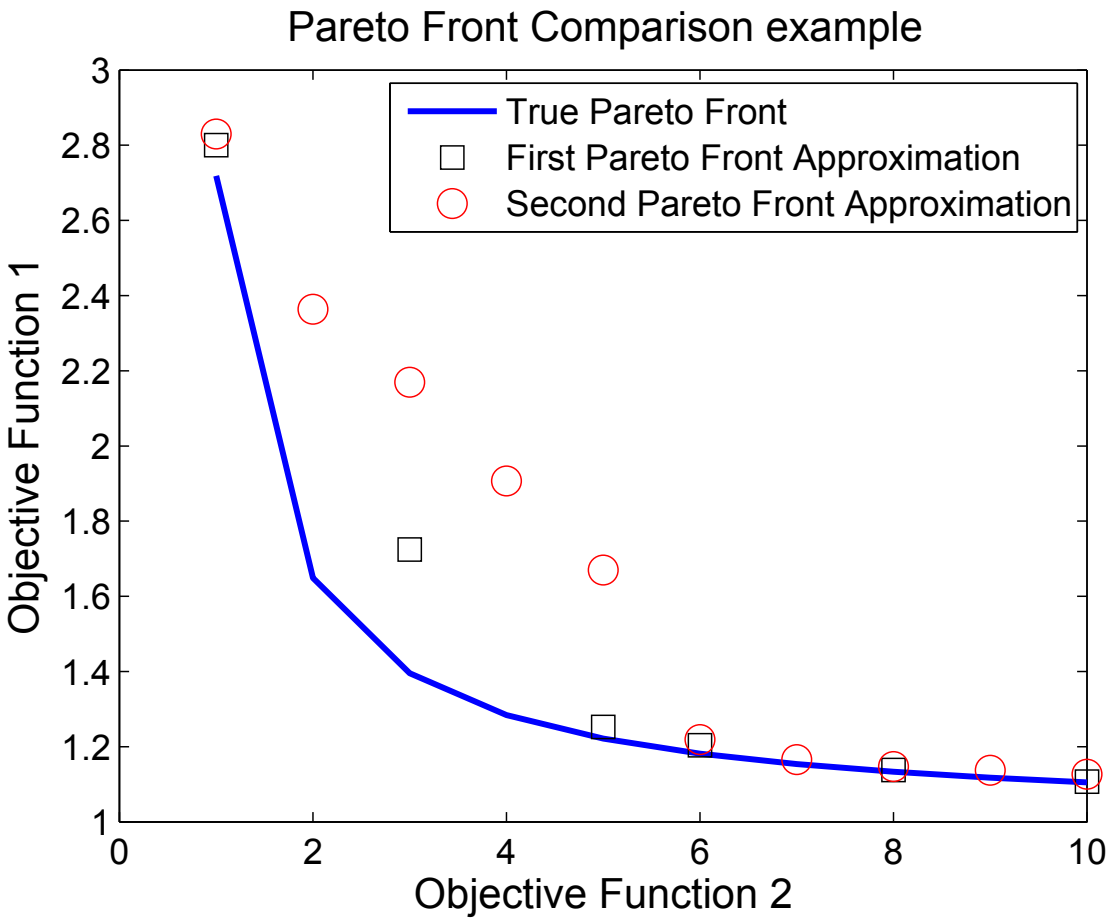
- $\vec{z}^1 \succ \vec{z}^2$ *($\vec{z}^1$ dominates $\vec{z}^2$) if $\vec{z}^1$ is not worse than $\vec{z}^2$ in any objective and is better in at least one objective*
- $\vec{z}^1 \succ\succ \vec{z}^2$ *($\vec{z}^1$ strictly dominates $\vec{z}^2$) if $\vec{z}^1$ is better than $\vec{z}^2$ in all objectives*
- $\vec{z}^1 \succeq \vec{z}^2$ *($\vec{z}^1$ weakly dominates $\vec{z}^2$) if $\vec{z}^1$ is not worse than $\vec{z}^2$ in any objective*
- $\vec{z}^1 \succeq_\epsilon \vec{z}^2$ *($\vec{z}^1$ epsilon dominates $\vec{z}^2$) if $\vec{z}^1$ is not worse than $\vec{z}^2$ by a factor of ϵ in any objective for a fixed $\epsilon > 0$*
- $\vec{z}^1 \parallel \vec{z}^2$ *($\vec{z}^1$ and $\vec{z}^2$ are incomparable to each other) if neither $\vec{z}^1$ weakly dominates $\vec{z}^2$ nor $\vec{z}^2$ weakly dominates $\vec{z}^1$*

With the outcome of a MOEA we are expecting to obtain an approximation as similar as possible (where this similarity concept will be deal with later in this work) to the Pareto set. This requires the formal definition of the outcome of a MOEA in terms of Pareto dominance, the approximation set (Hansen & Jaszkiewicz, 1998).

Definition 2.6.2 (Approximation set): *Let $A \subseteq Z$ be a set of objective vectors A is called and approximation set if any two members of A do not dominate each other: $\forall z^1, z^2 \in A : z^1 = z^2 \vee z^1 \parallel z^2$ The set of all approximation sets is defined as Ω*

Once the approximation set has been properly defined, the dominance relations presented in definition 2.6.1 can be extended to relations between different approximation sets, presented in definition 2.6.3

Figure 2.22: Comparison between different Pareto Front approximations regarding their quality



Definition 2.6.3 (Dominance relations applied to approximation sets): *Let $A_1, A_2 \in \Omega$ be two approximation sets The dominance relation $A_1 \succ A_2$ (A_1 dominates A_2) if every member of A_2 is dominated by at least one member of A_1 The remaining dominance relations presented in definition 2 6 1 ($\succ\succ, \succeq, \succeq_\epsilon, \|$) can be extended in a similar way*

The extension of dominance relations to approximation sets also allows the introduction of a new relation: *better*, represented as $A \triangleright B$ and presented in definition 2.6.4

Definition 2.6.4 (Better than approximation sets relation): *Given two approximation sets, $A, B \in \Omega$, $A \triangleright B$ (A is better than B) if A weakly dominates B and A and B are different approximation sets*

The presented definitions are summarized in tables 2.1 and 2.2. In the definitions of the relations a minimization of the objectives has been assumed (without loss of generality) and the epsilon dominance definition is based on its multiplicative terms (for the additive terms definition see (Zitzler et al., 2003), also covered as part of the proposal in section 5.2.1).

Table 2.1: Dominance relations between objective vectors

Relation	Representation	Definition
strictly dominates	$\vec{z}^1 \succ\succ \vec{z}^2$	$\forall i \in \{1, \dots, n\} z_i^1 < z_i^2$
dominates	$\vec{z}^1 \succ \vec{z}^2$	$\forall i \in \{1, \dots, n\} z_i^1 \leq z_i^2 \wedge \exists z_i^1 : z_i^1 < z_i^2$
weakly dominates	$\vec{z}^1 \succeq \vec{z}^2$	$\forall i \in \{1, \dots, n\} z_i^1 \leq z_i^2$
epsilon dominates	$\vec{z}^1 \succeq_\epsilon \vec{z}^2$	$\forall i \in \{1, \dots, n\} z_i^1 \leq \epsilon \bullet z_i^2, \epsilon > 0$
incomparable	$\vec{z}^1 \parallel \vec{z}^2$	$\vec{z}^1 \not\succeq \vec{z}^2 \wedge \vec{z}^2 \not\succeq \vec{z}^1$

Table 2.2: Dominance relations between approximation sets

Relation	Representation	Definition
strictly dominates	$A \succ\succ B$	$\forall z^2 \in B \exists z^1 \in A : z^1 \succ\succ z^2$
dominates	$A \succ B$	$\forall z^2 \in B \exists z^1 \in A : z^1 \succ z^2$
better	$A \rhd B$	$\forall z^2 \in B \exists z^1 \in A : z^1 \succeq z^2 \wedge A \neq B$
weakly dominates	$A \succeq B$	$\forall z^2 \in B \exists z^1 \in A : z^1 \succeq z^2$
epsilon dominates	$A \succeq_\epsilon B$	$\forall z^2 \in B \exists z^1 \in A : z^1 \succeq_\epsilon z^2$
incomparable	$A \parallel B$	$A \not\succeq B \wedge B \not\succeq A$

The objective of quality measures is to compare the outcomes of multiobjective algorithms in a quantitative manner. The simplest way to perform such a comparison would be according to the previous dominance relations, in order to determine whether an outcome is better than another given the analyzed relation. However, the final objective of quality measures is more precise and complex, trying to answer how much better an algorithm is with respect to another one and, in case none is better than the other, whether certain aspects can be considered superior. In this context, the definition of quality indicator arises

Definition 2.6.5 (Quality indicator): *An m-ary quality indicator I is a function $I : \Omega^m \rightarrow \Re$, which assigns each vector $(A_1, A_2, \dots, A_n)$ of m approximation sets a real value $I(A_1, A_2, \dots, A_n)$*

It must be noted that algorithm comparison, given their stochastic nature, implies several runs and the proper statistical testing over those runs, in order to obtain statistically significant results (Fonseca & Fleming, 1996; Grunert da Fonseca et al., 2001). Also, quality indicators need interpretation in order to obtain conclusions from their results. To provide this explanatory capability, we will introduce the comparison method definition.

Definition 2.6.6 (Comparison method): *Let A, B be two comparison sets, $\vec{I} = (I_1, I_2, \dots, I_k)$ a combination of quality indicators and $E : \Re^k x \Re^k \rightarrow false, true$ a Boolean function which takes 2 real vectors of length k as arguments If all indicators in $\vec{I}$ are unary, the comparison method $C_{\vec{I},E}$ defined by $\vec{I}$ and E is a boolean function of the form*

$$C_{\vec{I},E}(A, B) = E(\vec{I}(A), \vec{I}(B))$$

where $\vec{I}(A') = (I_1(A'), I_2(A'), \dots, I_k(A'))$ for $A' \in \Omega$ If $\vec{I}$ contains only binary indicators, he comparison method $C_{\vec{I},E}$ is defined as

$$C_{\vec{I},E}(A,B) = E(\vec{I}(A,B), \vec{I}(B,A))$$

where $\vec{I}(A',B') = (I_1(A',B'), I_2(A',B'), \ldots, I_k(A',B'))$ *for* $A', B' \in \Omega$

When the concepts of comparison methods and dominance relations are linked, two important definitions are required: compatibility and completeness.

Definition 2.6.7 (Compatibility): *Let* $\succ$ *be an arbitrary binary relation on approximation sets The comparison method* $C_{\vec{I},E}$ *is denoted as* $\succ$*-compatible if for any* $A, B \in \Omega$

$$\begin{cases} C_{\vec{I},E}(A,B) & \Rightarrow A \succ B \\ & or \\ C_{\vec{I},E}(A,B) & \Rightarrow B \succ A \end{cases}$$

Definition 2.6.8 (Completeness): *Let* $\succ$ *be an arbitrary binary relation on approximation sets The comparison method* $C_{\vec{I},E}$ *is denoted as* $\succ$*-complete if for any* $A, B \in \Omega$

$$\begin{cases} A \succ B & \Rightarrow C_{\vec{I},E}(A,B) \\ & or \\ B \succ A & \Rightarrow C_{\vec{I},E}(A,B) \end{cases}$$

Unary quality indicators, such as the hypervolume (Zitzler & Thiele, 1998), have been commonly used in order to establish the quality of an algorithm's outcome. In (Zitzler et al., 2003) the analysis of the performance of comparison methods based on unary quality indicators is carried out. This analysis is motivated by the hypotheses of being able to measure different aspects of the quality of an approximation set by different indicators in order to finally combine them to compare different outcomes. This analysis brings the following two theorems:

Theorem 2.6.1 *Suppose an optimization problem with* $n >= 2$ *objectives where the objective space is* $Z = \Re^n$ *Then, there exists no comparison method* $C_{\vec{I},E}$ *based on a finite combination* $\vec{I}$ *of unary quality indicators that is* $\succ$*-compatible and* $\succ$*-complete at the same time, i e,*

$$C_{\vec{I},E}(A,B) \Leftrightarrow A \succ B$$

for any approximation sets $A, B \in \Omega$

Theorem 2.6.1 leads to the formulation of the required number of unary quality indicators for a given number of objectives, presented in theorem 2.6.2

Theorem 2.6.2 *Suppose an optimization problem with* $n >= 2$ *objectives where the objective space is* $Z = \Re^n$ *Let* $\vec{I} = (I_1, I_2, \ldots, I_k)$ *be a combination of* k *unary quality indicators and* $E := (\forall 1 <= i <= k : I_i(\{\vec{z}^1\}) >= I_i(\vec{z}^2)$ *a Boolean function such that*

$$C_{\vec{I},E}(\{\vec{z}^1\}, \{\vec{z}^2\}) \Leftrightarrow \vec{z}^1 \succeq \vec{z}^2$$

for any pair of objective vectors $\vec{z}^1, \vec{z}^2 \in Z$ *Then, the number of indicators is greater than or equal to the number of objectives, i e ,* $k >= n$

These theorems provide an insight into the fact that the dimensionality reduction performed by unary quality indicators involves a loss of knowledge. This does not imply that these indicators are useless (particularly for the assessment of incomparable sets) but hampers their inferential power. These limitations do not exist, theoretically, for binary indicators. For instance, the binary epsilon indicator can be used to obtain a $\succ\succ$-complete and compatible comparison method, with the equation $I_\epsilon(A, B) < 1$.

2.6.2 Attainment functions

Section 2.6.1 has presented quality assessment from the perspective of quality indicators and comparison functions built upon them. This approach tries to deal with a reduction of the non-dominated Pareto Fronts which form the solutions of MOEAs and reduce the dimensionality to real values which are used for the statistical assessment based on the results over multiple runs. Attainment functions do not perform such a reduction, and handle directly the outcome of the algorithms. In (Grunert da Fonseca et al., 2001) the assessment of an optimizer performance in considered in terms of:

- Time taken to produce a solution with a given level of quality (run time)
- Quality of the solutions produced within a given time

The emphasis of these two performance considerations is the fact that, for stochastic optimizers (such as MOEAs) or deterministic optimizers under random conditions, both terms are random. In (Hoos & Stutzle, 1998) an estimation and analysis of the run-time distributions is proposed. In (Fonseca & Fleming, 1996) the study of the solution-quality distribution is suggested. The outcome of multiobjective optimization run was considered to be the set of non-dominated objective vectors evaluated during that run. Thus, in the single objective case, every run provides a single objective value, leading to univariate distributions. In the multiple objective case, every run provides either a single non-dominated vector per run (leading to multivariate distributions) or the general case which requires set distributions.

Attainment functions are based on the following definitions:

Definition 2.6.9 (Random non-dominated point set): *A random point set*

$$\chi = \{X_1, \ldots, X_M \in \Re^d : P(X_i <= X_j) = 0 i \neq j\},$$

where both the number of elements M and the elements X_j themselves are random and $P(0 \leq M \leq \inf) = 1$, is called a random non-dominated set (RNP-set)

Random Pareto-set approximations produced by stochastic multiobjective optimizers on d-objective problems are RNP-sets in $\Re^d$.

Definition 2.6.10 (Attained set): *The random set*

$$Y = \{y \in \Re^d | X_1 \leq y \vee X_2 \leq y \vee \ldots \vee X_M \leq y\} = \{y \in \Re^d | \chi \trianglelefteq y\}$$

is the set of all goals $y \in \Re^d$ attained by the RNP-set χ

It is key to state that the distributions of both random sets, χ and Y, are equivalent, meaning that a characterization of the distribution χ automatically provides a characterization of the distribution Y and vice versa.

Definition 2.6.11 (Attainment indicator): *Let $I\{\cdot\} = I_{\{\cdot\}}(z)$ denote the indicator function Then, the random variable $b_\chi(z) = I\{\chi \trianglelefteq z\}$ is called the attainment indicator of χ at goal $z \in \Re^d$*

Relating this indicator with the previously presented quality indicator definition (def. 2.6.5), as an infinite-dimensional quality indicator, it can be to construct a comparison method which is complete (def. 2.6.8) and compatible (def. 2.6.7) with respect to weak dominance.

Definition 2.6.12 (Attainment function): *The function $\alpha_\chi : \Re^d \mapsto [0,1]$ with*

$$\alpha_\chi(z) = P(b_\chi(z) = 1)$$

is called the attainment function of χ

In (Grunert da Fonseca et al., 2001) this attainment function was identified as the first-order measurement of the binary random field $b_\chi(z), z \in \Re^d\}$ derived from Y. This concept is related to random closed theory, particularly the so called *hitting function* or capacity functional (Cressie, 1992; Goutsias, 1998). This tool offers a useful description of the location of the distribution of Y (and, therefore, also of χ). The empirical counterpart of the attainment function $\alpha_\chi(\cdot)$ is defined as follows

Definition 2.6.13 (Empirical attainment function): *Let $b_1(z), \ldots, b_n(z)$ be n realizations of the attainment indicator $b_\chi(z), z \in \Re^d$ Then, the function defined as $\alpha_n : \Re^d \mapsto [0,1]$ with*

$$\alpha_n(z) = \frac{1}{n} \cdot \sum_{i=1}^{n} b_i(z)$$

is called the empirical attainment function (EAF) of χ

The realizations $b_1(z), \ldots, b_n(z)$ correspond to the n runs of the optimizer under study. While the theoretical attainment function is continuous, the EAF is a discontinuous function which exhibits transitions not only at the data points but also at other points, the coordinates of which are combinations of the coordinates of the data points (this is similar to multivariate empirical distribution functions (Justel et al., 1997)).

Since the EAF serves as an estimator for the theoretical attainment function $\alpha_\chi(z)$, the performance of an optimizer on a given optimization problem can be assessed via EAF estimates. It is remarkable that this assessment is performed in terms of *location* of the corresponding RNP-set distribution. A suitable, Smirnov-like statistical testing procedure based on two EAFs is applied in (Shaw et al., 1999). Rejecting the null hypotheses of equal attainment functions in a statistically significant way supports the conclusion that the optimizers under study exhibit different performance. However, If the null hypothesis cannot be rejected, the optimizers may still exhibit different performance, due to the statistical error involved, and the fact that the RNP-set distribution of a stochastic multiobjective optimizer is not completely characterized by the attainment function.

The attainment function, given its first-order moment nature, describes the distribution of the RNP-set χ in terms of location, but fails to address the dependent structure within the non-dominated elements of χ. For that purpose measures of second-order moment type are required. These measures allow the pairwise relationship between the elements of a random Pareto-set approximation χ to be studied.

Definition 2.6.14 (Second-order attainment function): *The function defined as* $\alpha_{\chi}^{(2)}(z_1, z_2) : \Re^d x \Re^d \mapsto [0, 1]$, *with*

$$\alpha_{\chi}^{(2)}(z_1, z_2) = P(b_{\chi}(z_1) = 1 \wedge b_{\chi}(z_2) = 1)$$

is called the second-order attainment function of χ

The second-order attainment function is the second, non-centred, moment measure of the binary random field $\{b_{\chi}(z), z \in Re^d\}$ derived from the attained set Y (Grunert da Fonseca et al., 2001). In random set theory terminology, the second-order attainment function would be called the covariance of the attained set Y ((Stoyan et al., 1995)). It expresses the probability of the same Pareto-set approximation χ simultaneously attaining two different goals $z_1, z_2 \in \Re^d$. The second-order attainment function is symmetric in its arguments, including all the information of the (first-order) attainment function, as

$$\alpha_{\chi}^{(2)}(z, z) = \alpha_{\chi}(z) \text{ for all } z \in \Re^d$$

and

$$\alpha_{\chi}^{(2)}(z_1, z_2) = \alpha_{\chi}^{(2)}(z_2, z_1) = \alpha_{\chi}(z) \text{ for all } z_1 \leq z_2 \in \Re^2$$

The empirical counterpart of this second-order attainment function is defined as follows:

Definition 2.6.15 (Second-order empirical attainment function): *Let* $b_1(z), \ldots, b_n(z)$ *be* n *realizations of the attainment indicator* $b_{\chi}(z), z \in \Re^d$ *Then, the function* $\alpha_n^{(2)} : \Re^d x \Re^d \mapsto [0, 1]$ *with*

$$\alpha_{\chi}^{(2)}(z_1, z_2) = \frac{1}{n} \cdot \sum_{i=1}^{n} b_i(z_1) \cdot b_i(z_2)$$

is called the second-order empirical attainment function of χ *(second-order EAF)*

The second-order EAF is a discontinuous function with the values $\alpha_{\chi}^{(2)}(z_1, z_2)$ representing the portion of optimization runs (Pareto-set approximations) which attained goals z_1 and z_2 simultaneously.

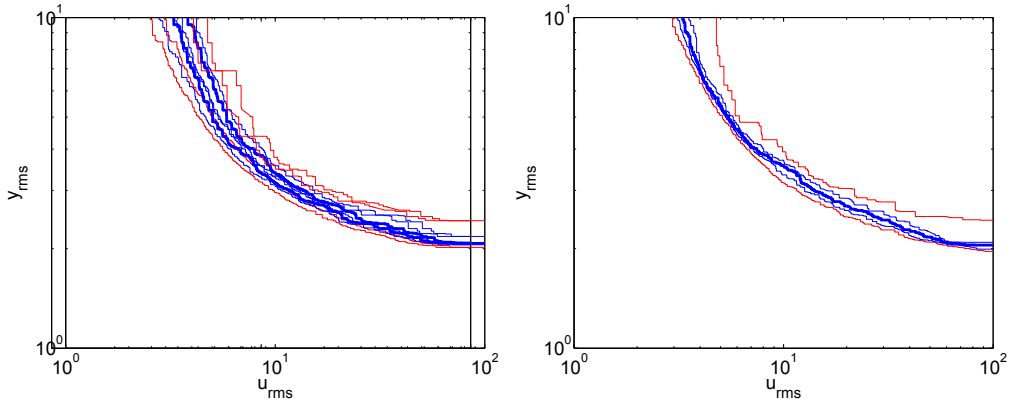
The optimizer's second-order behavior may be studied as well by the second, *centered*, moment measure of the binary field $\{b_{\chi}(z), z \in \Re^d\}$. This measure, according to random set theory literature, is referred to as the covariance function (Stoyan et al., 1995), as is defined as follows.

Definition 2.6.16 (Covariance function): *The function* $cov_{\chi} : \Re^d x \Re^d \mapsto [-0.25, 0.25]$ *with*

$$cov_{\chi}(z_1, z_2) = \alpha_{\chi}^{(2)}(z_1, z_2) - \alpha_{\chi}(z_1) \cdot \alpha_{\chi}(z_2)$$

is called the covariance function of χ

Figure 2.23: EAF examples over two different MOGA



(a) MOGA-A (b) MOGA-B

For each pair of goals $(z_1, z_2) \in \Re^d x \Re^d$ the value of the covariance function indicates the direction and the strength of the relationship between the two attainment indicators $b_\chi(z_1)$ and $b_\chi(z_2)$. A value of zero indicates no linear relationship between the elements, a positive value indicates that the attainment of goal z_1 tend to coincide with the attainment of goal z_2, while negative values indicate that the tendency to not attain both goals simultaneously. The maximum value of $cov_\chi(z_1, z_2)$ equals 0.25, and is reached by the variance function

$$var_\chi(z) = cov_\chi(z, z)$$

at all $z \in \Re^d$ where $\alpha_\chi(z) = 0.5$. An empirical counterpart of the covariance function is defined as follows.

Definition 2.6.17 (Empirical covariance function): *The function* $cov_n : \Re^d x \Re^d \mapsto [-0.25, 0.25]$ *with*

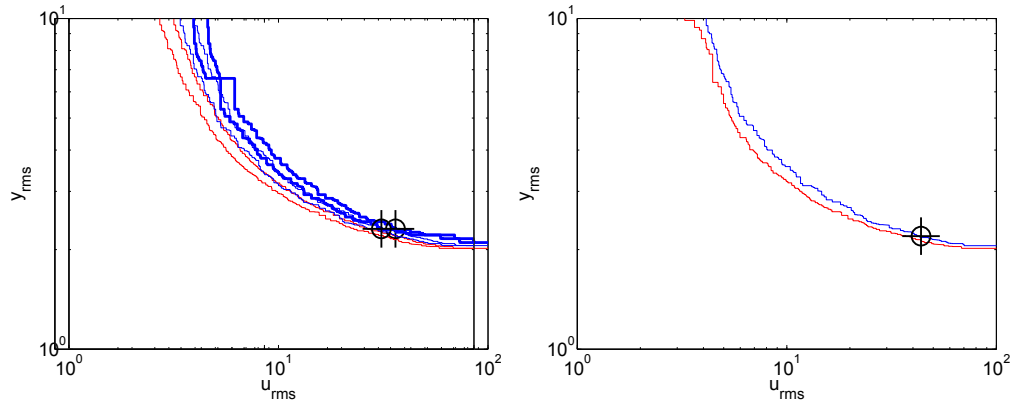
$$cov_n(z_1, z_2) = \alpha_n^{(2)}(z_1, z_2) - \alpha_n(z_1) \cdot \alpha_n(z_2)$$

is called the empirical covariance function of χ *(ECF)*

The visualization of attainment functions (particularly second-order moments) requires some workaround. In (Fonseca et al., 2005) this matter is presented, along with comparison examples to reflect the capabilities of first and second-order moments. One of the examples included presents the comparison between two MOGAs, one without sharing or mating restriction (MOGA-A) and one with sharing and mating restriction in the variable domain (MOGA-B) (Fonseca & Fleming, 1995). Contours for EAF are drawn (from left to right) at the $\epsilon-$, $0.25-$, $0.5-$, $0.75-$, and $(1-\epsilon)-$ levels, for arbitrarily small positive epsilon. Figure 2.23 (Fonseca et al., 2005) shows these results for the proposed example, where figure 2.24a represents the results for the MOGA-A and figure 2.24b for MOGA-B

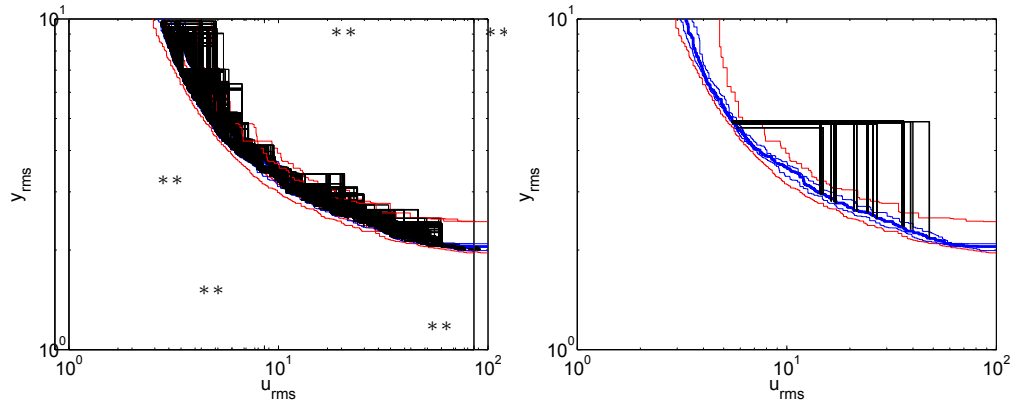
The visualization of second-order EAF is more difficult than that of the first order EAF, as it is defined in $\Re^{2d}$. With two objectives, a useful workaround consists in fixing one goal $z^* \in \Re^2$ and depicting the contours of the marginal function $\alpha_n^2(z^*, z)$ defined over all $z \in \Re^2$, at given levels. Figure 2.24 (Fonseca et al., 2005) shows the contour of the second-order EAF at levels ϵ, 0.25 and 0.5. Figure 2.25a shows these three levels, due to

Figure 2.24: Contour plot examples of marginal second-order EAF with two different z^* goals



(a) Higher z^* goal results (b) Lower z^* goal results

Figure 2.25: Pairs (z_1, z_2) showing covariance values above or below a certain threshold



(a) Covariance value greater than 0 21 (b) Covariance value lower than -0 21

the fact that $\alpha_n(z^*) < 0.75$ in this case, leading to $\alpha_n^{(2)}(z, z^*) \forall z \in \Re^2$. Figure 2.25b has a further downwards goal z^*, leading to the loss of another contour level.

Finally, the visualization of the empirical covariance function also requires a special technique. In figure 2.25 (Fonseca et al., 2005) the pairs of goals which exhibit a covariance value above or below a certain threshold are indicated in the objective space by a solid bracket beginning at one goal and ending at the other, with the contours of the first-order EAF plotted as a reference in the background. These figures show objectives which are particularly likely and unlikely to be attained simultaneously.

The experimental results for this example in (Fonseca et al., 2005) showed that, while the EAFs of the two different MOGAS did not show statistically significant differences, the results over their second order EAFs leaded to the rejection of the null hypotheses (equality) at an 0.05 alpha level. This shows that the first-order EAF does not fully characterize the outcome of the optimizers, such that, as pointed out previously, the lack of rejection

of the null hypotheses (even with second-order moments) does not completely exclude the possibility of different performance between the two compared optimizers.

2.7 Approaches to stopping criteria in Multiobjective evolutionary algorithms

The mathematical background for convergence in MOEAs is still a work in progress in the research community. Several works have dealt with the convergence of MOEAs to the true Pareto front in a finite number of function evaluations. (Rudolph & Agapie, 2000) and (Rudolph, 2001) proved that such convergence could be achieved by MOEAs with positive elitism and positive variation kernel. This results were extended to $t \rightarrow \inf$ by (Hanne, 1999; Laumanns, 2003; Laumanns et al., 2002). Local optimality of solutions can be guaranteed with the use of quadratic programming methods (Deb et al., 2007; Wanner et al., 2006), which can establish a mathematical convergence criteria. However, the quality of the set of solutions is not guaranteed.

The most recent approaches start with (Deb & Jain, 2002) and the proposal of running performance metrics for convergence and diversity of solutions in the course of the optimization run, which lead to the algorithm stop when convergence was observed. This proposal was followed by (Roudenko & Schoenauer, 2004), where a survey of possible alternatives was carried out, selecting a stagnation criteria based on the maximum crowding distance, thus focusing on NSGA-II algorithm, which uses it as its selection criterion (Deb et al., 2002b).

2.7.1 Online Convergence Detection algorithm (OCD)

Online Convergence Detection algorithm (OCD) (Trautmann et al., 2009; Wagner et al., 2009) establishes two different stopping criteria based on values obtained from a set of quality indicators. Both tests are applied to a given window. The first of these tests focuses on the variance analysis, trying to determine whether this value is too small regarding the generations in the window to continue the optimization. The second criterion analyzes the trend in the values of the performance indicators, focusing on whether there is a significant improvement or not. OCD determines the stop of the optimizer if any of the two tests are triggered in two consecutive generations, providing as its output the final generation and the criterion which triggered the stop.

OCD requires a certain number of arguments to be established. The variance limit *VarLimit* is the parameter which determines the desired approximation accuracy in single-objective optimization. For a given window size of *nPreGen* generations, if the standard deviation of the indicator values falls significantly below $\sqrt{VarLimit}$, the criterion is triggered. In (Wagner et al., 2009), a value for this parameter is empirically proposed: $\sqrt{VarLimit} = 10^{-3}$. An important feature of OCD is that the adaptation of this parameter is not required due to different expected ranges in the objective functions values, since and internal normalization to the interval $[1, 2]^d$ is carried out. The statistical significance level α for the different statistical tests is proposed to be set to 0.05 (as a standard value) or 0.01 (as a more conservative value). The *MaxGen* parameter expresses the maximum value for runtime resources (which might be set according to generations or function evaluations).

The number and types of desired performance indicators (PI) can be selected to evaluate the solution quality regarding the requirements of the user. The quality indicators integrated

in OCD are chosen according to (Knowles et al., 2005), which proposes the hypervolume (Zitzler & Thiele, 1998), the additive ϵ (Zitzler et al., 2003) and the R2 indicator (Hansen & Jaszkiewicz, 1998). Additive ϵ and R2 indicators require a reference test (since they are binary quality indicators). This set is the output at the current generation, as is used as a reference for all the previous *nPreGen* values. For the hypervolume indicator (a unary quality indicator), the difference between the current generation and all the previous *nPreGen* values is used.

The variance criterion performed by OCD checks whether the resulting *nPreGen* vectors of n indicator values fall below the predefined *VarLimit* threshold. For that purpose, a χ^2 test is used (Sheskin, 2000). The global significance level α is adjusted due to the multiplicity of the test problem using a Bonferoni correction (Dudoit & van der Laan, 2008). This implies a $\frac{\alpha}{n}$ for each PI variance test result.

The regression criterion uses a linear regression analysis without intercept and a respective $t-test$ on the estimated regression coefficient $\widehat{\beta}$. This analysis requires a preprocessing step where the values of PI_j are standardized, which means that they are linearly transformed to mean zero and standard deviation one, so the regression is performed on all indicators at once.

Due to a termination in cases where the p value is higher than α, a more conservative α leads to an earlier termination (as pointed out by intuition). Since there are several tests performed with the same α value, it would be interesting to perform a combination and analysis of the error which this α level implies. However, a combination of the α level in both tests cannot be performed with respect to multiple test theory (Dudoit & van der Laan, 2008). Thus, it is important to notice that the objective is not to control the α error, but instead to find reasonable critical values of the test statistics.

The description of the whole algorithm is presented in algorithm description 1, and the statistical tests used in 2 (χ^2 test) and 3 ($t-test$).

2.7.2 MGBM stopping criterion

The MGBM criterion (Martí et al., 2007, 2009) uses Roudenko's work (Roudenko & Schoenauer, 2004) as its base to establish a stopping criterion according to a self-defined improvement indicator and the use of a Kalman filter's (Welch & Bishop, 1995) output. The authors establish the requirement of any stopping criteria to address two different issues: first of all the need for a measure of the improvement obtained by a given solution after an iteration of the MOEA, and secondly a mechanism which may keep track of those measurements over time in order to decide whether the execution of the MOEA should be stopped or not.

The argument behind the requirement for special performance indicators lies in the original application of quality indicators to compare Pareto fronts to optimal solutions, in order to determine the quality of an optimizer's outcome. They may be reformulated to compare two solutions on different generations of a given run of an optimizer (such as performed by OCD (section 2.7.1), but that involves a high computational cost in the computation of the indicator (since they were designed for offline processing). This leads to the creation of a new indicator from scratch, inherently designed for the performance measurement purpose. In MGBM, this responsibility lies in the Mutual Dominance Rate (MDR) indicator.

MDR is a metric based on set of non-dominated solutions of two consecutive generations, P_t^* and P_{t-1}^*. A help function is used in the formulation of the indicator, the function $\triangle(A, B)$

Algorithm 1 Algorithm for Online Convergence Detection

Require: $VarLimit$ {maximum variance limit}
$nPreGen$ {number of preceding generations for comparison}
α {significance level of the tests}
$MaxGen$ {maximum number of generations}
$(PI_1, \ldots, PI_n)$ {vector of performance indicators. Default: $(HV, \epsilon, R2)$}
Ensure: $\{MaxGen, Chi2, Reg\}$ {criterion which terminates the MOEA}
Ensure: i {generation in which the criterion holds}
i=0
repeat
i=i+1
Compute d objective Pareto front PF_i of ith MOEA generation
$\vec{lb} = min(\vec{lb} \cup PF_i)$
$\vec{ub} = max(\vec{ub} \cup PF_i)$
if $i > nPreGen$ **then**
$PF_i^{'} = 1 + (PF_i - \vec{lb})/(\vec{ub} - \vec{lb})$
for all $k \in \{i - nPreGen, \ldots, i-1\}$ **do**
$PF_k^{'} = 1 + (PF_k - \vec{lb})/(\vec{ub} - \vec{lb}$
end for
for all $j \in \{1, \ldots, n\}$ **do**
$PI_{j,i} = (PI_j(PF_{i-nPreGen}^{'}, PF_i^{'}, 1, 2.1), \ldots, (PI_j(PF_{i-1}^{'}, PF_i^{'}, 1, 2.1)))$
{Compute PI_j for $PF_{i-nPreGen}^{'}, \ldots, PF_{i-1}^{'}$ using}
{$PF_i^{'}$as reference set, 1 as ideal and 2.1 as nadir point}
$pChi2(j, i) = callChi2(\vec{PI}_{j,i}, VarLimit)$ {pvalue of the χ^2 test}
end for
$pReg(i) = callReg(\vec{PI}_{1,i}, \ldots, \vec{PI}_{n,i})$ {pvalue of the t-test on the generation's}
{effect on the $PI_{j,i}$}
end if
until $\forall j \in \{1, \ldots, n\} : (pChi2(j, 1) \leq \alpha/n) \wedge (pChi2(j, i-1) \leq \alpha/n)$
$\vee\ (pReg(i) > \alpha) \wedge (pReg(i-1) > \alpha)$
$\vee\ i = MaxGen$
Terminate MOEA
return $\{MaxGen, Chi2, Reg\}, i$

Algorithm 2 Chi2: One-sided χ^2 variance test for
H_0 : var(PI) $\geq$ $VarLimit$ vs H_1 var(PI) $<$ $VarLimit$

Require: $\vec{PI}$ {vector of performance indicator values}
$VarLimit$ {Variance limit}
Ensure: p {Resultant p-value of the test}
$N = length(\vec{PI} - 1$ {determine degrees of freedom}
$Chi = [var(\vec{PI} \cdot N]/VarLimit$ {computes test statistic}
$p = \chi^2(Chi, N)$ {Lookup χ^2 distribution with N degrees of freedom}
return p

Algorithm 3 Reg: Two-sided $t-test$ on the significance of the linear trend
$H_0 : \beta = 0$ vs $H_1 : \beta \neq 0$

Require: $\vec{PI}_j, j = \{1, \dots, n\}$ {vector of performance indicator values}
Ensure: p {Resultant p-value of the test}
  $N = length(\bigcup_{j=1}^{n} \vec{PI}_j) - 1$
  for all $j \in \{1, \dots, n$ **do**
    $\vec{PI}'_j = (\vec{PI}_j - mean(\vec{PI}_j))/std(\vec{PI}_j$ {standardise}
  end for
  $\vec{Y} := concatenate(PI'_1, \vec{\dots}, PI'_n)$ {row vector of all $\vec{PI}'_j$}
  $\vec{X} := (1, \dots, length(\vec{PI}_1), \dots, 1, \dots, length(\vec{PI}_n))$
    {row vector of generations corresponding to each $\vec{PI}_j$}
  $\widehat{\beta} = (\vec{X} * \vec{X}^T)^{-1} * \vec{X} * \vec{Y}^T$ {linear regression without intercept}
  $\epsilon = \vec{Y} - \vec{X} * \widehat{\beta}$ {compute residuals}
  $s^2 = (\epsilon * \epsilon^T)/N$ {mean squared error of regression}
  $t = \frac{\widehat{\beta}}{\sqrt{s^2(\vec{X} * \vec{X^T})^{-1}}}$ {compute test statistic}
  $p = 2 \cdot min(t_N(t), 1 - t_N(t))$
    {look up p value from t distribution with N degrees of freedom}
  return p

that returns the set of elements of A that are dominated by at least one element of B. This operator is formally presented in equation 2.8

$$C = \triangle(A, B), \text{ such that } \forall x \in C, x \in A, \exists y \in B \text{ with } y \prec x \tag{2.8}$$

The MDR progress indicator $I_{mdr}(P^*_t, P^*_{t-1}) \in [-1, 1]$ contrasts how many non-dominated individuals of iteration t dominate the non-dominated individuals of the previous generation $t-1$ and viceversa, and is represented in equation 2.9

$$I_{mdr}(P^*_t, P^*_{t-1}) = \frac{|\triangle(P^*_{t-1}, P^*_t)|}{|P^*_{t-1}|} - \frac{|\triangle(P^*_t, P^*_{t-1})|}{|P^*_t|} \tag{2.9}$$

where $|A|$ is the number of elements in A.

The interpretation of the values of the MDR indicator is the following: $I_{mdr} = 1$ indicates that the population from iteration t is completely better than its precedent one, $I_{mdr} = 0$ indicates that no progress has been performed in the generation, while $I_{mdr} = -1$ indicates the complete degradation in the solution quality of generation t.

The complexity of determining the non-dominated individuals at every iteration can be computationally expensive. However, this is usually one of the steps in the running cycle of most MOEAs, and thus can be embedded into them. Having this local Pareto-optimal fronts P^*_t and P^*_{t-1} the complexity order of I_{mdr} is $O(M \cdot |P^*_t| \cdot |P^*_{t-1}|)$.

The MGBM criterion bases its evidence gathering process on Kalman filters (Welch & Bishop, 1995). This implies the assumption that the noise present in the measured progress indicator is uncorrelated between consecutive iterations. Furthermore, the estimated value of the progress indicator (and its associated covariance, used internally by the filter) follows a Markov process (Bharucha-Reid, 1960), which implies that the outcome of each iteration is only dependent on the previous one.

The Kalman filter assumes a dynamic model given by equation 2.10

$$x_t = Ax_{t-1} + Bu_t + w_t \tag{2.10}$$

where u_t is an optional control input and the random variable w_t $N(0, Q)$ represents the process noise. The measurement process is modeled by equation 2.11

$$z_t = Hx_t + v_t \tag{2.11}$$

where H relates the real state of the process x_t to the measurement z_t and v_t $N(0, R)$ is the measurement noise.

Kalman filter, therefore, provides the means to estimate the state of a dynamic system from a series of incomplete and noisy measurements. Its state is represented by two variables: $\widehat{x_t}$ which is the state estimate at time t, and P_t, which is the error covariance matrix (a measure over the estimated accuracy of the current state estimate).

The Kalman filter operates according to two different phases: prediction and update. Prediction phase makes an *a priori* estimation of the future estate according to current information (state, covariance matrix and model). The computation of a priori variables can be performed with equations 2.12, 2.13.

$$\widehat{x}_t^- = A\widehat{x}_{t-1}^- + Bu_t \tag{2.12}$$

$$P_t^- = AP_{t-1}^- A^T + Q \tag{2.13}$$

The Kalman filter computes at every step the Kalman gain, K, value, which will be used during the update phase to integrate the real measured values into the filter state. This gain can be calculated with equation 2.14

$$K_t = \frac{P_t^- H^T}{HP_t^- H^T + R} \tag{2.14}$$

Once this Kalman gain has been obtained and the new measurement is received, its value is injected into the filter with equations 2.15, 2.16

$$\widehat{x}_t = x_t^- + K_t(z_t - Hx_t^-) \tag{2.15}$$

$$P_t = (I - K_t H)P_t^- \tag{2.16}$$

In MGBM criterion, this filter is used according to its simplest possible model, which, according to equation 2.12, established a constant movement without any control input ($A = 1$ and $B = 0$). Also the prediction error in the dynamic model is disregarded ($Q = 0$ in equation 2.13 and also regarding the variable w_t in the model equation (2.10). These assumptions present a model according to equation 2.17, where the measurement model is presented in equation 2.18.

$$\widehat{I}_t^- = \widehat{I}_{t-1}^- \tag{2.17}$$

$$z_t = I_{mdr}(P_t^*, P_{t-1}^*) \tag{2.18}$$

Extending these simplifications to the remaining equations, the error covariance also remains constant (equation 2.19), the Kalman gain can be computed with equation 2.20 and finally the a posteriori estimate can be obtained with equation 2.21

$$P_t^- = P_{t-1} \tag{2.19}$$

$$K_t = \frac{P_t^-}{P_t^- + R} \tag{2.20}$$

$$\widehat{I}_t = \widehat{I}_t^- + K_t(z_t - \widehat{I}_t^-) \tag{2.21}$$

The idea of the criterion, following these equations, is to stop the MOEA once no further progress is being detected ($\widehat{I}_t = 0$). The constant R provides a mean to control the inertial response of the filter, providing a faster or slower reaction to changes. Different configuration values are suggested for this parameter, determining an empirically obtained value of 0.1.

2.8 Time series segmentation and polygonal approximation

Time series domain involves a set of different concepts and procedures to understand its importance and the available techniques. The objectives of this introductory section are to present those concepts in a simple way, leading the reader to easily understand the growing importance of time series, the different representation issues and how they have been faced by the available approaches, introduce the air traffic control domain in the context of time series, and finally outline the main aspects of the technique developed in this work.

Time series are sequences of data vectors, containing each of these vectors a timestamp as one of its values. Typical examples can be found in genetic research(Yeang & Jaakkola, 2005), financial(Taylor, 2008), medicine(Clifford et al., 2006) manufacturing(Ge & Smyth, 2001) or tourism(Yu & Schwartz, 2006) applications. Several processes can be defined regarding time series, such as their analysis (in order to extract different meaningful characteristics or patterns from them, which can be used by additional processes) or forecasting (the development of models in order to predict future values).

The importance of time series has grown exponentially in recent years, due to the explosion in the application of collection and storage technologies, generating huge amounts of data to be processed. In the financial domain, a clear example is the tracking of stock prices, being constantly updated in different markets all over the world(Gionis & Mannila, 2005). The processing of these massive amounts of data requires an approximate representation of the information that can be more efficiently and effectively handled (as the analysis of every time point is usually not necessary nor practical, and can even be unaffordable). Time series segmentation is a tool presented in order to resolve this issue, by means of reducing the data dimension with appropriate models for representation and approximation.

To achieve that dimensionality reduction, segmentation processes may use different high level representations, such as Fourier Transforms(Brockwell & Davis, 2009), Wavelets(Percival & Walden, 2006), Symbolic Mappings(Balzanella et al., 2010) or the approach, recently explored by the data mining community(Gionis & Mannila, 2005; Keogh et al., 2003; Liu et al., 2008), which will be covered in this paper: Piecewise Linear Representation (PLR, also named Piecewise Linear Approximation, PLA). We have centered

the scope of our proposal in this representation due to its wide application to different domains. This representation can be treated both as a final result itself and / or as the basis for different additional processes, such as fast similarity search(Keogh et al., 2001). The extended use of this particular technique may be caused by its simplicity and ease of use: PLR segmentation is based on the approximation of a Time Series T with length n by means of a set of K segments (where $K << n$), approximating each of these segments by a linear model.

A segmentation technique, in general, is responsible of the division of a time series into a certain number of segments (ideally, as few as possible) and the approximation of the data in each segment by a certain simple function. This introduces several interesting issues, such as the measurement of the quality of a certain segmentation result and the consideration of the implied cost to obtain that quality (remembering that the purpose of the whole schema is to perform a dimensionality reduction over the original data). Different classifications can be performed over segmentation techniques along with the high level representation used, being a capital one, regarding their applicability to different processes, their online or offline nature. Offline segmentation algorithms may use the whole data from the time series to obtain their segment approximation(Keogh et al., 2003), whereas online algorithms perform their segmentation based only on the available data of an incomplete data series(Liu et al., 2008). This nature usually has an impact on the complexity and accuracy of the resulting algorithm (offline algorithms benefit from their complete knowledge of the time series to obtain a more accurate segmentation, while their computational complexity is more impacted by the size of the considered time series).

Regardless of the concrete technique applied (or according to it, considering that this fact may determine the concrete approximation used), representation of the available information is the key to obtain effective and efficient segmentation results(Zhu et al., 2010). Time series may be affected by a series of factors, such as large quantity of measurements and the presence of severe noise in them which may prevent those achievements. Thus, dealing with those handicaps is an extremely important issue for segmentation processes. These factors are especially relevant in time series exhibiting sensor data or video tracking information(Machos et al., 2004).

2.8.1 Formalization

The objective of a segmentation process is to divide a data sequence into a series of segments and approximate these segments with a simple function. In the case of study in this work, PLR, those segments are approximated with piecewise linear models.

The segmentation process can be seen as a search over the time series measurements trying to obtain the structure of segments that minimize (or maximize) a certain quality function. Considering each measurement represented as $\vec{x}_k$ for a time series T, the segmentation process is formalized in (2.22)

$$\begin{aligned} T = \{\vec{x}_k\}, S(T) = \{B_m\}, B_m = \vec{x}_j, j \in [k_{min}, \dots, k_{max}] \rightarrow \\ \rightarrow \begin{matrix} min \\ max \end{matrix} \quad f_{quality}(\{B_m\}) \end{aligned} \tag{2.22}$$

being $S(T)$ the result of segmentation according to the criteria in the given function $f_{quality}$, which is minimized or maximized according to the given requirements. The best

possible solution for this process could be obtained by considering every possible segment obtained from the different $\vec{x}_k$ measurements of the time series and deciding the output value according to the summation of the function error values for those segments. Equivalently, this can be seen as a search over the different possible measurements which divide the trajectory into different segments. Unfortunately, these search processes are computationally unaffordable, leading to different segmentation techniques which apply different heuristics.

The traditional criteria to determine the quality of a segmentation process(Keogh et al., 2003; Liu et al., 2008) are the following:

1. Minimizing the overall representation error (*total_error*)
2. Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)
3. Minimizing the number of segments so that the total representation error does not exceed *total_error*

where *total_error* and *max_segment_error* are user defined parameters for the algorithm.

2.8.2 Time series segmentation algorithms overview

Time series domain has proposed different algorithms according to different heuristics to solve the segmentation issue. The three main approaches are based on sling windows for online approaches, constructive approaches based on the iterative join of different segments to provide the final segmentation output and, finally, destructive approaches based on the split of an initial segment (representing the whole data series) to meet some quality thresholds.

Sliding window(Appel & Brandt, 1983) is an online algorithm based on building growing windows from the beginning of the time series until a certain user boundary is exceeded by the result of an error function, leading to the creation of a new segment at that measurement and the restart of the process. Several improvements have been performed over this basic version, such as the Incremental Sliding Window(Liu et al., 2008), or the different complementary approaches(Vullings et al., 1997). It is also important to notice that the sliding window algorithm is reported to give pathological results under certain circumstances(Agrawal et al., 1993).

Top Down algorithm(Keogh et al., 2003) is an offline process based on finding the best splitting point (understanding by this that measurement which divides the trajectory into the two segments with the lowest added errors) recursively, until all the resulting segments have an error value bellow a user defined boundary. The Top Down algorithm is applied in a wide variety of domains and fields, being also known by different names(Douglas & Peucker, 1973; Duda & Hart, 1973). As in the case of the sliding window, there are numerous improvements to the basic top down algorithm. Alternative approaches(Park et al., 1999) perform different initializations based on valleys and peaks

Bottom up algorithm(Keogh et al., 2003) is an offline process complementary to Top Down, where the time series is initially divided into every possible segment (composed of two measurements) and finds the best possible segment fusion afterwards (understanding by this the fusion which obtains the segment with the lowest error) until any possible fusion obtains a segment having an error above a user defined boundary. The bottom up algorithm, as well, has spread to different fields and research areas using different names, such as the computer graphics domain and decimation methods(Heckbert & Garland, 1997).

2.8.3 Polygonal approximation algorithms overview

Polygonal approximation techniques for segmentation are a particular case of time series segmentation where the data analyzed is a closed curve and the timestamp is simply a relative ordering among the points. These data streams can be formalized according to equation 2.23, which defines the components of a given curve, where x_i and y_i are the plane coordinates of the point and i is the point's number.

$$t = \{\vec{p_i}\}, \vec{p_i} = (x_i, y_i, i), i = 1, \dots, n \tag{2.23}$$

From the definition of the input data included in equation 2.23, the definition of a segmentation process may be formalized with equation 2.24, where each B_m is one of the resultant segments, composed of a set of $\vec{p_i}$ points. Dominant points are those at the extremes of each of these segments, k_{min} and k_{max}, which delimit them.

$$S(t) = \{B_m\}, B_m = \{\vec{p_i}\}, i = k_{min}, \dots, k_{max}\ m \in [1, \dots, n-1] \tag{2.24}$$

Approximation algorithms can be divided into sequential, split and merge and heuristic search approaches. Sequential approaches are constructive methods based on a given local search over the current data, trying to obtain, at each step, a new segment division (where the length of these segments is sequentially increased) which satisfies a certain criterion. Examples of the used criteria may be finding the longest possible segments (Sklansky & Gonzalez, 1980) or a combination of finding the longest possible segments with the minimum possible error (Ray & Ray, 1992). Split and merge approaches perform an initial segmentation over the given time series and afterwards start an iterative process to merge the initial segments until a certain criterion is met. According to their definition, these approaches have to deal with two different issues: the initial segmentation procedure and the merging criterion. An example of these techniques is (Ramer, 1972), which performs an initial boundary segmentation, followed by a sequence of steps where the segment is split at the point with the furthest distance from the corresponding segment unless the approximation error is lower or equal than an specified error tolerance.

Heuristic search approaches are based on the development of heuristic methods in order to avoid the exhaustive search of the optimal *dominant points* for the given curve (which is a process with an exponential complexity). Different techniques may be used for this purpose, such as dynamic programming (Dunham, 1986; Sato, 1992) or several metaheuristics, among them solutions based on genetic algorithms (Goldberg et al., 1989; Pal et al., 1998; Tsai, 2006; Yin, 1999, 1998), which will be highlighted in this work due to their relationship with the proposed solution. The idea proposed by these works is to codify the time series as a chromosome with n genes, corresponding each of these genes to one of the points in the original data. If the gene value is a "1", it is considered a *dominant point*, and the algorithm tries to find the ideal codification of the chromosome according to a fitness function which evaluates the quality of the given codified segmentation in the chromosome.

To provide an insight into the importance of dominant points for these approaches, we will describe two specific polygonal approximation approaches: Teh and Chin algorithm (Teh & Chin, 2002) and Marji and Siy algorithm (Marji & Siy, 2003).

Teh and Chin algorithm (Teh & Chin, 2002) is based on the concept of the *region of support* (Langridge, 1972): this concept states that each boundary point of a closed curve

must have its own view of the curve, being dominant points those which have a meaningful view of the curve which blocks the view of other non-dominant points.

In (Teh & Chin, 2002) the proposal is based on the difficulty of determining the curvature of a digital curve. The functions to determine discrete curvature are named *measures of significance* (Rosenberg, 1972). Three different measures of significance are used: the k cosine measure, the k curvature measure and the 1 curvature measure. The algorithm starts with the calculation of the region of support for a given point p_i. This calculation is performed determining the length of the chord joining the points p_{i-k} and p_{i+k} (l_{ik}, shown in equation 2.25) and the perpendicular distance of the points contained in the chord to their respective ones in the original data, d_{ik}. This process is continued until the value of the length of the chord stops growing or until the mean distance starts growing (represented in equation 2.26).

$$l_{ik} = \overline{|p_{i-k}p_{i+k}|} \tag{2.25}$$

$$\begin{cases} \frac{d_{ik}}{l_{ik}} \geq \frac{d_{i,k+1}}{l_{i,k+1}}, & \text{if } d_{ik} > 0 \\ \frac{d_{ik}}{l_{ik}} \leq \frac{d_{i,k+1}}{l_{i,k+1}}, & \text{if } d_{ik} < 0 \end{cases} \tag{2.26}$$

The second step of the algorithm calculates the three measures of significance. Finally, according to the previous data, dominant points are calculated suppressing non-maximal points from the previous sets. This is performed following an iterative process which sequentially filters the points according to their measure of significance value. This process changes depending on the concrete measure used.

Marji and Siy algorithm (Marji & Siy, 2003) relies on the concept of *support arms*. This implies that the region of support is not used to calculate a significance measure of the boundary points, but instead compute the strength of the end points of their calculated regions of support, both in clockwise and counterclockwise directions. This strength is determined by the frequency of their choice.

To determine both support arms, the function shown in equation 2.27 is maximized, where L_{jk} is the length of the segment joining points p_j and p_k and E_{jk} is the sum of the squared perpendicular distances of the points contained between p_j and p_k to that segment. This is performed increasing iteratively the length of the region until that increase makes the function obtain a lower value. When that happens, the previous end point is considered the support point. Variable k has an initial value of $j+2$ or $j-2$, depending on which support arm is being calculated.

$$F = L_{jk} - E_{jk} \tag{2.27}$$

The algorithm proceeds calculating the support arms of all points, and determining their classification as dominant depending on the classification of the points surrounding them, their distance to the segment delimited by the dominant points immediate to their left and right and the possible overlapping of regions of support.

2.9 Conclusions and analysis

This fundamentals chapter has proposed the required framework for the thesis proposal, according to the following principles: overview of problem classification and applicability of

metaheuristic approaches, description and definition of these approaches, according to their two complementary functions (diversification and intensification), presentation of evolutionary approaches, according to their main characteristics, with a special focus on stopping criteria and multi-objective approaches, being finally followed with a more in depth analysis of quality assessment and several available multi-objective stooping criteria approaches. Also, a coverage of the application domain has been included, introducing the segmentation issue with a special focus on the approach followed, Piecewise Linear Representation.

The analysis of metaheuristic applicability shows that there is not an immediate application relationship between tractability of a problem and the requirement for metaheuristic approaches to solve it (as covered in section 2.2). This resembles the automatic assignation usually made of NP-hard problems to metaheuristic approaches and P problems to exact algorithms. Also, a factor which must be taken into account for the application to real problems is the design cost of a particularized exact algorithm to solve a particular tractable problem, comparing it with the representation and objective function design (sections 2.4.1 and 2.4.3) which are required for the application of a general purpose metaheuristic such as evolutionary algorithms.

This supposed ease of application usually assigned to evolutionary approaches (where, ideally, the only domain information required is introduced in the representation and fitness assignation steps) must also be reviewed with care. This application guideline poses every of the steps covered in figure 2.6 as a *black box*, where independent improvements and research can be performed , with a strict relationship to the representation used, as seen in section 2.4. On the one hand, this approach has allowed the independent focus on each of these processes and the independent improvement over each of them (particularly noticeable in the transformation operators, mutation and crossover, as covered, respectively, in sections 2.4.4 and 2.4.5). Also, it simplifies the application of these algorithms for practitioners, being closely related to their current extended use.

On the other hand, however, this independent overview of the different steps of evolutionary algorithms (and metaheuristics in general, as introduced in section 2.3) introduces a series of handicaps. The introduction of domain knowledge at different additional steps during the algorithm development has been proved to be beneficial (particularly for initialization procedures, as analyzed in section 2.4.2). Also, the inclusion of an external decision maker may prove beneficial, integrating cycles of automatic optimization and decision making (as seen in the progressive techniques overviewed in section 2.5.1). Finally, there may be uncovered interactions between different choices for these different *black boxes*, which may require a detailed analysis. This is the case, for instance, of population diversity and stopping criteria (shown in the classification of approaches in section 2.4.7).

Stopping criteria have been one of those steps receiving the least amount of research interest, being a topic related mostly to practical applications. In most approaches (particularly visible when quality comparisons are involved), this has been resolved based on static exhaustion budgets (set according to the difficulty of the particular problem or test set). According to this criterion, the comparison was based on the quality of the obtained solutions. In many real application cases (where the cost of the fitness function computation clearly determines the complexity of the algorithm) the decision maker or algorithm designer usually assumes the role of this stopping criterion (since there is no *a-priori* budget to be fulfilled).

The stopping criteria step is closely related to quality assessment (covered in section 2.6). which has continuously been the objective of a great amount of interest and research,

since it is a crucial step for the performance measurement of new approaches and alternatives (being this performance usually measured omparatively according to previously available approaches). This relationship is clearly visible in the newly developed approaches to stopping criteria in multi-objective optimization (covered in section 2.7). The study of quality assessment technique reveals that attainment functions (section 2.6.2) are yet to find applicability to stopping criteria approaches, generally based on quality indicators (section 2.6.1), probably due to their more extended application for quality measurement (usually based on statistical testing over hypervolume indicator values) and the inherent complexity of the technique (even though some of the difficulties related to its application to a set of different executions would disappear in its application as part of a stopping criterion).

Covering the specific stopping criteria approaches overview in this state of the art, OCD 2.7.1 displays a robust approach to stopping criteria based on previous studies regarding the applicability of different quality indicators and techniques to quality assessment. Several issues, may be mentioned, however: the establishment of the different configuration and design parameters are not strictly related to the evolutionary theory, but rather to statistical considerations, making them hard to comprehend and modify for the standard researcher. Also, as pointed out in the algorithm description, this robustness comes at a certain price.

First of all, due to the different tests performed, no possible analysis of the error introduced in the α parameter can be performed. Typical values are suggested for this parameter, based on empirical results to find reasonable critical values for the test statistics. This approach to the α parameter, even though it obtains remarkable results, somehow contrasts with the robustness and highly theoretical support for the formulation of the different steps of the algorithm. The effect of the resources window must also be taken into account. Another important handicap inherited from this robustness based approach (according to the quality assessment analysis followed) is the computational cost: for every new algorithm generation, a complete re-normalization of the different Pareto fronts stored must be performed, the indicators recomputed with respect to the new last front and the criterion tests reapplied. This considerations may be particularly important for some of the indicator presenting a higher computational cost, such as hypervolume.

MGBM stopping criterion 2.7.2 follows the complementary path to OCD: to avoid computationally expensive stopping criterion calculation, it presents a brand new progress indicator to cope with the comparison of the different Pareto fronts, and guides the assessment according to this indicator. The mutual dominance indicator (MDR) resembles the archiving procedure introduced in SPEA2, basically considering, as it name points out, the mutual dominance between the elements of the two considered Pareto fronts. Also, according to this performance based approach, guidelines to the introduction of this indicator as part of the traditional evolutionary cycle are provided. The arising question here is whether this indicator provides enough information to guide the stopping assessment properly or not. Also, even though it is proposed as a binary indicator following the quality assessment guidelines available, its properties, particularly regarding compatibility and completeness (definitions 2.6.7 and 2.6.8).

Covering the information processing performed, a remarkable choice is the suppression of one of the noise sources (Kalman filtering provides one for the underlying considered model and one for the measuring process). While this choice is justified in the algorithm detailed analysis, this introduces the consideration of whether a different model should not be chosen for this task (which did not include the suppressed noise source), or even a different processing

schema introduced. Once again the input parameters may not be easily understandable for a researcher focused on evolutionary computation, which may not be familiar with linear estimation techniques, being also faced with the presentation of suggested values according to empirical results.

Regarding the available segmentation techniques, one of the arising difficulties are the different domains which they are applied to, which has led to different research lines which in some cases have led to similar algorithms. An analysis over these different domains also highlights the cost of dealing with specific domain issues, which imply a high degree of development for the different heuristics developed. Also, as noted in section 2.8, a segmentation is a dimensionality reduction, and the results of that reduction are traditionally measured in terms of its quality, disregarding the cost of that reduction. A possible explanation for this fact is that the presented heuristic approaches lack a proper direct control mechanism over that cost.

3

An initial non-evolutionary approach to the application domain: HLRA

> *He found insanity no excuse, however, for irrational behavior Some men were blind, others had poor tempers Still others heard voices It was all the same, in the end A man was defined not by his flaws, but by how he overcame them*

Brandon Sanderson, *Mistborn:The Well of Ascension*, 2007

This chapter will present the Air Traffic Control domain as an application domain of Piece Linear Representation and segmentation techniques. This domain presents some very interesting properties which increase the difficulties of traditional heuristic techniques applied in segmentation approaches. Even though the solution currently presented is not evolutionary based, some of the applied techniques are closely related. The problem's nature is analyzed according to some of the representation theory introduced in sections 2.4.3 (formalization of fitness functions) and 2.5.1 (multi-objective evolutionary approaches), leading to its formalization as a multiobjective problem. The solution comparison resorts to quality indicators, which were covered in section 2.6.1, and the performance assessment of the introduced technique, in general, relies in quality assessment developments for evolutionary algorithms (section 2.6). Also, this development opens future lines regarding evolutionary solutions for a further application of the multiobjective nature highlighted in this chapter, which are one of the focuses of this thesis, and will be developed in chapters 6 and 7. The main reference works for this chapter are (Guerrero et al., 2011, 2010; Guerrero & García, 2009)

3.1 Introduction

Time series segmentation, along with some of its most representative approaches, has been covered in section 2.8. A paradigmatic time series domain is Air Traffic Control (ATC), which analyzes the data coming from sensors measuring the position of aircraft, which is recorded for offline validation, resulting in time series usually named *opportunity traffic*. This *opportunity traffic* information is the only available experimental data in this domain.

In this section, ATC will be used as a source for opportunity traffic time series in order to perform PLR segmentation over them. The particular importance of these time series

is related to the domain activities: ATC is a critical area related with safety, requiring strict validation in real conditions (Garcia et al., 2009), being this one of the previously mentioned domains where the amount of data has gone under an exponential growth (in this case due to the increase in the number of passengers and flights). This has led to the need of automation processes in order to help the work of human operators (Baud et al., 2009). These automation procedures can be basically divided into two different processes: the required online tracking of the aircraft (along with the decisions required according to this information) and the offline validation of sensor data processors. The evaluation task is usually separated into two sub-processes, segmentation (Guerrero & García, 2008; Guerrero et al., 2010b), showing a slightly different meaning to the one previously introduced, as it only covers the division of the time series into a series of segments, and reconstruction (Garcia et al., 2009), which covers the approximation of the segments which the trajectory was divided into. Artificial intelligence techniques have been applied for different purposes, such as establishing a formalization of the domain theory and its associated validation process (West & McCluskey, 2001). Considering it from the segmentation point of view, opportunity traffic provides very interesting time series due to the difficulties which segmentation processes have to face in them. These difficulties, along with the characteristics of data measurements, may include reformulations of the quality functions used to measure the accuracy of a segmentation result (due to the high noise in the measurement values and the knowledge of the motion models which the aircraft may perform).

This chapter reviews the performance of traditional segmentation techniques and proposes a new approach for these particularly difficult domains to deal with for PLR segmentation: noisy domains with a large number of measurements. The technique presented is built according to established segmentation design characteristics, also discussing the treatment which these design characteristics have received in the available algorithms. The presented approach will lead to the introduction of the Hybrid Local Residue Analysis (HLRA) segmentation technique, particularizing its results for the ATC domain. The introduction of quality measures is also required in order to cope with the noisy data and the multiobjective nature of the problem solutions, which, along with the proper statistical testing, will be used to test the relevance of the obtained results over a dataset containing opportunity traffic trajectories from the ATC domain.

3.2 Segmentation issues in the ATC domain

The traditional PLR segmentation techniques exhibit a series of problems and issues in domains with high noise and very long series. These domains are particularly interesting when there is available information about the introduced noise (or accurate estimations of its value), being a clear example of this fact the measurements obtained by sensor devices (having an individual model for their measuring errors), which have in the ATC domain one of its most representative examples. Typical sensor devices are radar (Skolnik, 2008), GPS (Farrell & Barth, 1999), or inertial (Groves, 2008). This data is often processed by intermediate data fusion architectures (Liggins et al., 2008) (these architectures will be briefly covered, according to their possible application to multiobjective evolutionary algorithms stopping criteria, in section 5.2.3). Thus, the introduction to these domains will be performed by the description of the particular ATC domain, in order to be able to build the argumentation leading to the proposed technique.

The basic data in the ATC domain are the trajectories recorded from flying commercial aircraft, containing sensor plots with the following components: stereographic projections of their x and y components (which are a representation in a common reference frame of the different radar measurements), covariance matrix (representation of the noise introduced by the positioning system: radar, GPS, multilateration, etc) and detection time. This domain also allows us to exploit some related knowledge due to the fact, as has already been pointed out in references such as (Garcia et al., 2009) that the movement models (MM's) of commercial flights have a certain uniformity in their values (meaning that they tend to follow certain MM's smoothly, without abrupt changes in the position values). This prevents the application of approaches to detect abrupt changes, such as the one exposed in (Zhu et al., 2010), based on the identification of those changes (named *feature points*).

In this domain, the models followed by an aircraft can be usually simplified into three different possibilities achieving remarkable results (Garcia et al., 2009): uniform, turn and acceleration MM's (which might be reduced even to only two, considering that a turn is only a transversal acceleration MM). An important consideration is the length of the maneuvers when we compare them to the uniform segments of the trajectory. If we consider that in a whole time series q measurements were recorded while the aircraft was performing some kind of maneuver and p measurements were recorded while the aircraft was performing a uniform MM, for the vast majority of trajectories, $p >>> q$. These trajectories are performed in airways areas, the most common situation in the available airspace. When the plane approaches a terminal in the surroundings of airports, it gets into terminal maneuvering areas (TMA), where this rule does not apply. To illustrate these differences we have included into the considered dataset racetracks examples, the trajectories performed by aircraft during the landing procedures, which we will analyze later.

Therefore, the right identification of the uniform segments becomes the key factor in this domain (which involves the difficulty of being able to differentiate the effects of the noise from those due to the start or end of a maneuver). An effective PLR segmentation technique should be able to adequately identify those long uniform segments accurately. There is very valuable information which algorithms must seek to introduce. This information includes noise and maneuvers data. The noise introduced in the time series' values is caused, as explained, by different measuring devices, usually external, such as radars (Skolnik, 2008) or automatic dependent surveillance (ADS) systems based on GPS (Williams, 2009). Usually the segmenting algorithm is provided this information by a covariance matrix under Gaussian assumptions, not requiring it to know or apply special noise considerations depending on the measuring device. The additional important source of information involves the minimum and maximum length of the maneuvers the aircraft may take (which is specially delimited when handling commercial air traffic). This data can provide us with configuration parameters for our algorithms, in order to adjust them to the kind of traffic they will be dealing with.

According to the analysis presented in this section, PLR segmentation techniques will have to deal in the ATC domain time series with three difficulties: the noise introduced by the measurement device, the large number of measurements which compose each trajectory performed by an aircraft and the proper segmentation of the long uniform segments which these aircraft exhibit.

3.3 Time series segmentation techniques

The objective of a segmentation process is to divide a data sequence into a series of segments and approximate these segments with a simple function. In the case of study in this work, PLR, those segments are approximated with piecewise linear models.

The segmentation process can be seen as a search over the time series measurements trying to obtain the structure of segments that minimize (or maximize) a certain quality function. Considering each measurement represented as $\vec{x}_k$ for a time series T, the segmentation process is formalized in equation (3.1)

$$T = \{\vec{x}_k\}, S(T) = \{B_m\}, B_m = \vec{x}_j, j \in [k_{min}, \dots, k_{max}] \rightarrow \\ \rightarrow \begin{matrix} min \\ max \end{matrix} \quad f_{quality}(\{B_m\}) \tag{3.1}$$

where $S(T)$ is the result of segmentation according to the criteria in the given function $f_{quality}$, which is minimized or maximized according to the given requirements, and B_m is a concrete segment from the solution (which covers the points between k_{min} and k_{max} boundaries). The best possible solution for this process could be obtained by considering every possible segment obtained from the different $\vec{x}_k$ measurements of the time series and deciding the output value according to the summation of the function error values for those segments. Equivalently, this can be seen as a search over the different possible measurements which divide the trajectory into different segments. Unfortunately, these search processes are computationally unaffordable, leading to different segmentation techniques which apply different heuristics.

The traditional criteria to determine the quality of a segmentation process (Keogh et al., 2003; Liu et al., 2008) are the following:

1. Minimizing the overall representation error (*total_error*)
2. Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)
3. Minimizing the number of segments so that the total representation error does not exceed *total_error*

where *total_error* and *max_segment_error* are user defined parameters for the algorithm.

The previous criteria highlight the fact that, instead of a single quality function, these processes usually have to minimize (or maximize) a set of different error functions jointly (typically an error function measuring the distance to the original time series, for example an Euclidean distance, and a different one measuring the cost of that error, for example the number of segments used for the segmentation), changing this approach into a multiobjective optimization problem (MOOP) (Ehrgott, 2005), formally represented by equation (3.2). Additionally, sets of restrictions over the quality functions may be set.

$$T = \{\vec{x}_k\}, S(T) = \{B_m\}, B_m = \vec{x}_j, j \in [k_{min}, \dots, k_{max}] \rightarrow \\ \rightarrow \begin{matrix} min \\ max \end{matrix} \quad \{f_{q1}(\{B_m\}), \dots, f_{q1}(\{B_m\})\} \tag{3.2}$$

Following the formalization in equation (3.2), we may introduce the three quality criteria presented for the PLR problem obtaining equation (3.3).

$$B_m = \vec{x}_j, j \in [k_{min}, \dots, k_{max}], m \in [1, \dots, seg_{num}] \rightarrow min\{d(S(T)), T), seg_{num}\} \text{ such that } \begin{cases} d(S(T), T) \leq total\_error \\ \forall m, d(f_{ap}(B_m), B_m) \leq max\_segment\_error \end{cases} \quad (3.3)$$

where $d(P, Q)$ is a distance error function between series P and Q, $f_{ap}(B)$ is the approximation function result over series B (in PLR the resulting line which approximates the data in series B), and seg_{num} is the number of segments obtained by the applied segmentation algorithm.

Comparing the three different formalizations for the segmentation process presented following the theory in section 2.4.3 (objective function definition for evolutionary algorithms), the evolution is stated from a single objective, unconstrained optimization function (initially presented in its general form in equation 2.2 and particularized for current domain in equation 3.1) to a multi-objective formulation introducing the different criteria stated (presented in equation 2.4, along with a set of general constrains, and the current domain adaptation without such constrains in equation 3.2), finally leading to the multiobjective constrained optimization formulation (where constrained optimization was presented in its single-objective version in equation 2.3, in its multiobjective version in equation 2.4 and particularized for the segmentation issue in equation 3.3).

The reader may notice that minimizing the number of segments seems to be a key factor in the quality of the segmentation process (as it appears in two out of the three criteria). Even so, most capital references on this topic, such as (Keogh et al., 2003), even though they state the three previous criteria, base their quality comparisons on only one factor: *total_error*. Only recently the number of segments is beginning to be compared as a performance metric over the quality of a segmentation process, in references such as (Liu et al., 2008).

The lack of attention to a performance metric which is, at the same time, stated to be a very important one, can be explained by looking at the design of traditional algorithms (which will be covered in the next subsections) and the absence in them of mechanisms to actually control that the number of segments is kept to an allowable minimum. Their approach is based on equation (3.1), using a leading quality function based only on the approach error compared to the original time series. The segmentation approach proposed in this chapter will take into account that value not only as a quality comparison value, but also as a design consideration. This is especially important for noisy domains, since considering only the representation error leads to oversegmentation in the trajectories, due to the algorithm's excessive focus on the position changes caused by the noise.

An obvious determinant factor not yet commented is the computational complexity of the segmentation process. In general, segmentation processes are required to have low computational complexity (or at least a scalable one), either in online (due to the real-time requirements) or offline (due to the huge amounts of data required) processing.

Along with the differentiation between online and offline algorithms, the linear segmentation process can be divided, as well, into two different approaches: linear interpolation and linear regression. The former uses the equation of a straight line given two points which belong to it (using the initial and end points of the segments) to obtain the approximation segment. This produces a segmentation of the time series with continuous piecewise lines.

On the other hand, linear regression, as its own name indicates, uses a regression line to approximate all the points belonging to the segment with a criterion of minimum residual error, producing a set of disjointed lines. The overall error obtained by a linear regression is always less than or equal to the one obtained with a linear approximation, which, along with the low computational complexity it involves, are some of the reasons for its usual choice (Keogh et al., 2003; Liu et al., 2008). There are additional difficulties faced by linear interpolation approaches in noisy domains, due to their high sensitiveness to the position of the initial and final measurements of the segment, which can redefine the interpolated segment completely. According to this analysis, we will introduce linear regression into the applied techniques presented in this chapter. Linear approximation, on the other hand, is required for polygonal approximation, so it will be the technique used in chapters 6 and 7.

A final decision regarding a general segmentation algorithm is whether the final segments will be continuous (meaning that the end measurement of segment i is the beginning measurement of segment $i + 1$) or discontinuous (each measurement belongs only to one segment). Some algorithms are more sensitive to this choice than others, being bottom up approaches the most affected by it (since a discontinuous approach introduces limitations in the possible sizes of the output segments). The algorithms presented in this chapter will use a continuous approach, in order to prevent possible limitations introduced by discontinuous segments.

An overview of the three traditional time series segmentation techniques which will be used in current chapter has been presented in section 2.8.2. Their implementation will be based in the pseudocodes provided in (Keogh et al., 2003). Relating the algorithm descriptions to current domain characteristics, particularly the noisy data, it becomes specially relevant the fact that sliding window (Appel & Brandt, 1983) is considered to obtain a best relative performance on noisy data (Keogh et al., 2003) and that alternative implementations of the Top Down algorithm, such as (Park et al., 1999) initializations based on valleys and peaks, are on the other hand, reported to perform poorly on noisy datasets.

3.4 The Hybrid Local Residue Analysis technique

In the presentation of segmentation techniques (section 3.3), the multiobjective nature of segmentation algorithms and the importance of the number of segments was introduced (equations 3.2 and 3.3). Even so, these techniques do not provide any explicit mechanism to deal with this performance metric, which explains the lack of coverage this parameter has in most of the available literature. This fact, which is important in any general domain, is even more so in the ATC domain. The reason for this importance is that, as was explained in section 3.2 of this chapter, typical ATC time series consist of very long uniform segments causing that, if our segmentation technique outputs a large number of segments, it probably means that we are oversegmenting, using that information to cover the position changes introduced by the noise in the measurements (which, in terms of storage, is a waste of capacity, and difficulties the processing of the output for a different range of processes, such as reconstruction (Garcia et al., 2009)).

This introduces a different quality factor to the presented domain: there is a requirement not only for a number of segments as low as possible, but also for the concentration of those segments on the maneuver sections of the analyzed time series. Generally, multi-objective formulations contrast some performance measure (in this case, the representation quality in

terms of an error value) versus some cost measure (in this case, the number of segments). The capital difference in current proposal is the fact that the comparison of performance versus cost measures is not sufficient to assess the result of the algorithm, since the noise may lead to misleading results, and thus there is requirement for a more complete analysis of the cost's investment. These considerations will be taken into account for the performance comparison introduced in section 3.6.1.

The general idea of the proposed approach, the Hybrid Local Residue Analysis (HLRA) technique, is to analyze each measurement of the trajectory according to a surrounding window and assign a classification value to it (local classification according to a residue value). This classification determines if the measurement is considered to belong to a uniform MM or non-uniform MM. Adjacent measurements sharing the same classification are considered to belong to the same segment. Once the whole time series has been classified following this approach, those segments which were classified as belonging to a non-uniform MM are segmentated according to the bottom up algorithm (hybrid segmentation schema). The segmentation positions obtained this way are relative to the beginning of the non-uniform segments, which require to be corrected to their respective positions in the complete time series in order to be added to the final solution. Figures 3.1 and 3.2 present an overview of the two phases of this process, while figure 3.3 shows an example over a turn trajectory. This example shows that the first phase of the algorithm, by the use of the local classification information, is able to accurately segmentate the time series data where the aircraft was performing a uniform MM, while those sections where a non-uniform MM was being performed are handled afterwards and segmentated by the general bottom-up algorithm.

Offline processing allows the use of information both from past and future values of the time series. Introducing this fact into a local representation, the information will be restricted to a certain local segment around the measurement which is to be classified. These intervals are centered on that measurement, but the boundaries for them can be expressed either in number of measurements (equation 3.4) or timestamp values (equation 3.5).

$$S_j^i = \{\vec{x}_k^i\}, k \in [j-p, j+p], p \in [j-1, N-j] \tag{3.4}$$

$$S_j^i = \{\vec{x}_k^i\}, t_k^i \in [t_j^i - m, t_j^i + m], m \in [t_j^i - t_1^i, t_N^i - t_j^i] \tag{3.5}$$

where S_j^i is a given segment from the trajectory centered on measurement j, N is the number of measurements contained in the time series, p is the sample window size and determines the possible boundaries for a given segment according to its number of measurements (from measurement 1 to measurement N) and m is the time window size, which determines those same possible boundaries according to their timestamp. Once the required window around current measurement has been properly chosen, a function is applied to that segment in order to obtain its classification. This general classification function $F(\vec{x}_j^i)$, using measurement boundaries, is represented in (3.6)

$$F(\vec{x}_j^i) = F(\vec{x}_j^i | T^i) \rightarrow F(\vec{x}_j^i | S_j^i) = F_p(\vec{x}_{j-p}^i, \ldots, \vec{x}_j^i, \ldots, \vec{x}_{j+p}^i) \tag{3.6}$$

From this formulation of the problem we can already see some of the choices available: how to choose the segments (according to equations (3.4) or (3.5)), which classification function to apply in eq. (3.6) and how to perform the final segment synthesis. An example

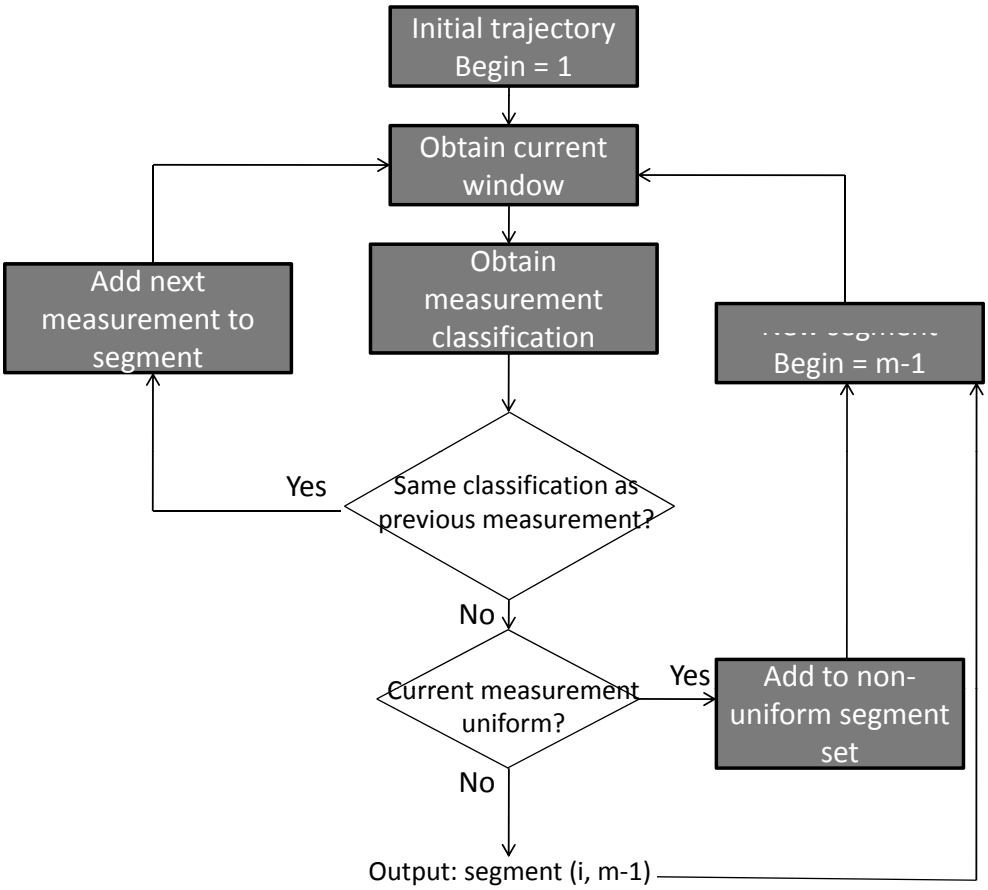


Figure 3.1: Hybrid Local Residue Analysis first phase overview

of the segmentation issue according to the local classification formulation is presented in figure 3.4.

The segment boundaries are defined by the domain knowledge. As exposed in the domain section, this knowledge is usually in the form of average duration (in time) of typical maneuvers, so the algorithm will resort to equation (3.5), setting a value for m adjusted to the half time duration of the longest possible maneuver. Once the analysis window has been fixed, the classification function will be based on a Best Linear Unbiased Estimator (BLUE) (Henderson, 1975) residue value (domain transformation), in order to introduce noise information in the uniform segment detection, and an automatic threshold choosing technique to determine the final classification over that value.

3.4.1 Introducing noise information: the BLUE residue

The first phase of our algorithm covers the process where we must synthesize an attribute from our input data to represent each of the trajectory's measurements in a transformed domain and choose the appropriate thresholds in that domain to effectively differentiate those which belong to our model from those which do not do so.

The transformation function decision is crucial. A BLUE residue value will be used, where

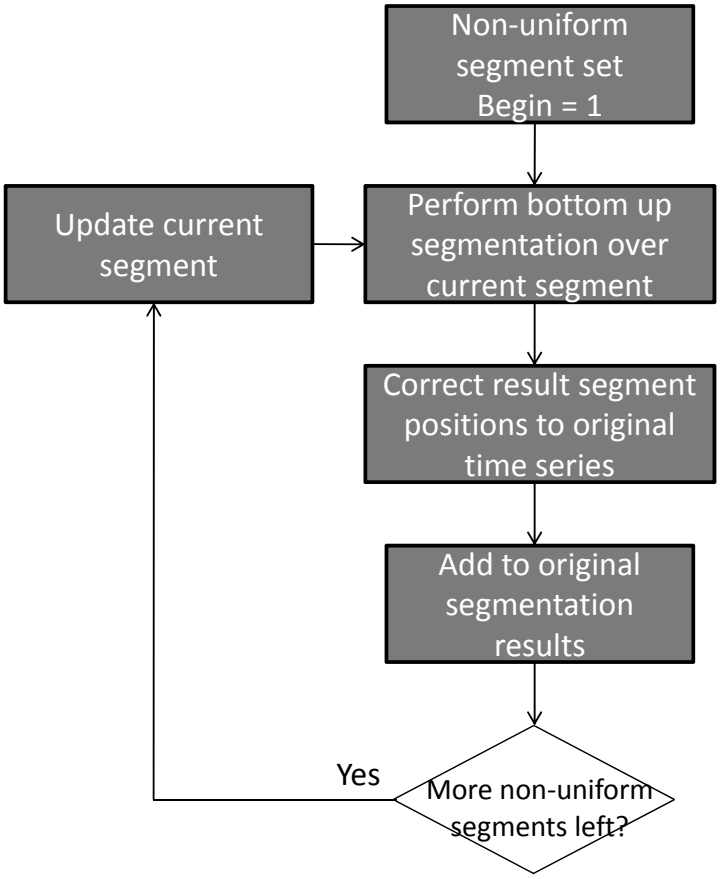


Figure 3.2: Hybrid Local Residue Analysis second phase overview

we will be able to introduce the noise information by means of a covariance matrix R_k. The assumed linear model is presented in equation (3.7)

$$\vec{x}_m(k) = \begin{bmatrix} x_m(k) \\ y_m(k) \end{bmatrix} = \begin{bmatrix} 1 & t_k & 0 & 0 \\ 0 & 0 & 1 & t_k \end{bmatrix} \begin{bmatrix} x_0 \\ vx_0 \\ y_0 \\ vy_0 \end{bmatrix} + \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix} = H(t_k)\vec{\theta} + \vec{n}(k) \tag{3.7}$$

The first component $H(t_k)\vec{\theta}$ represents the ideal estimated parameters for a uniform segment (initial position and velocity). The best estimator of these parameters with minimum squared weighted residual is introduced in eq. (3.8). The noise information is introduced in (3.8) in the form of its covariance matrix, R_k. Then, with estimator $\vec{\theta}$ the interpolated positions for the x and y components of the points can be calculated with eq. (3.9). Finally, with the previous values, the normalized BLUE residue can be obtained with eq. (3.10).

$$\left\langle \vec{\theta} \right\rangle = \begin{bmatrix} \langle x_0 \rangle \\ \langle vx_0 \rangle \\ \langle y_0 \rangle \\ \langle vy_0 \rangle \end{bmatrix} = (\sum_k H(t_k)^T R_k^{-1} H(t_k))^{-1} \sum_k H(t_k)^T R_k^{-1} \vec{x}_m(k) \tag{3.8}$$

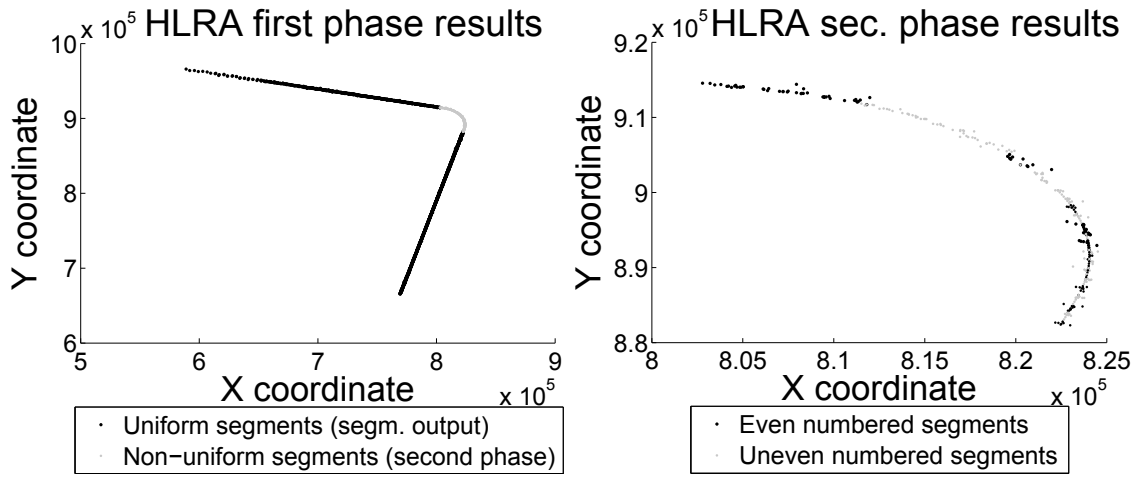


Figure 3.3: Example of the HLRA's results over a sample turn trajectory

$$x_{int}(k) = \langle x_0 \rangle + \langle vx_0 \rangle \, k \qquad y_{int}(k) = \langle y_0 \rangle + \langle vu_0 \rangle \, k \tag{3.9}$$

$$res = \frac{1}{kmax - kmin + 1} \sum_{k=kmin}^{kmax} \begin{pmatrix} x(k) - x_{int}(k) & y(k) - y_{int}(k) \end{pmatrix} R_k^{-1} \begin{pmatrix} x(k) - x_{int}(k) \\ y(k) - y_{int}(k) \end{pmatrix} \tag{3.10}$$

where $x(k)$, $y(k)$ are the sensor measurements values, R_k is the covariance matrix (associated to the sensor) and $x_{int}(k), y_{int}(k)$ are interpolated values using equation (3.9).

The threshold choosing technique is closely related to the domain transformation, involving how we determine if a measurement belongs to the model or not. The choice for this parameter will be detailed in the next section.

3.4.2 Threshold choosing technique

The threshold choice involves determining the boundary above which transformed measurements will be considered as unknown. Figure 3.5 shows an example of a possible choice over the presented transformed domain.

According to previous considerations, the objective is to classify the measurements belonging to a uniform MM correctly, with a special attention regarding the limits where the aircraft's MM changes to a different one. Graphically over figure 3.5, that implies getting the straight line as low as possible, leaving only the central section over it (where the maneuver takes place, making its residue value high enough to get over our threshold).

The presented residue value in eq. (3.10) follows a Chi-squared probability distribution function (pdf) normalized by its degrees of freedom, n. n is given by twice the number of 2D measurements contained in the interval minus the dimension of P (P=4 in our uniform segments, as we are imposing four linear restrictions). For a valid segment residual, *res*

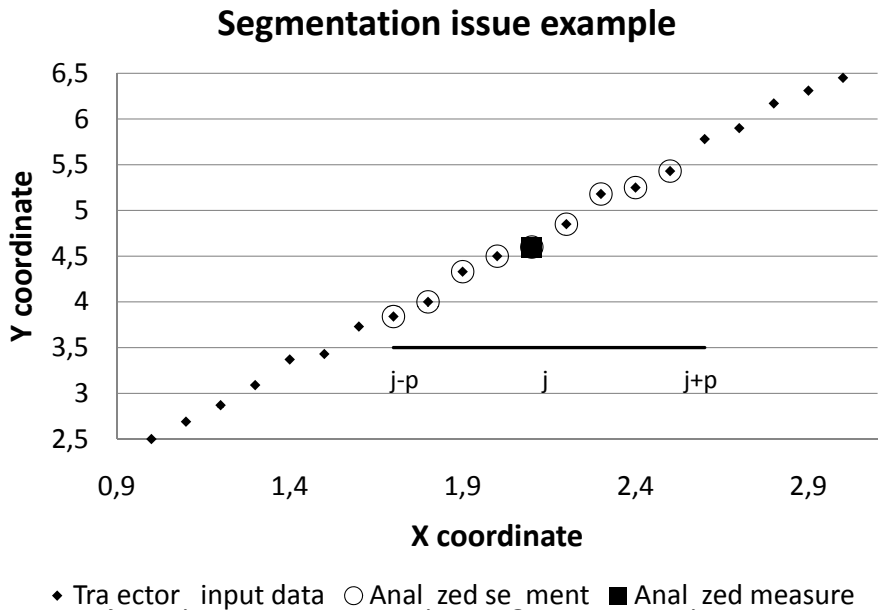


Figure 3.4: Local approach segmentation overview

behaves with distribution $\frac{1}{kmax-kmin+1}\chi^2_{2(kmax-kmin+1)-P}$, which has the mean and variance detailed in eq. (3.11).

$$\mu = 2 - \frac{P}{kmax - kmin + 1} \quad \sigma^2 = \frac{4}{kmax - kmin + 1} - \frac{2P}{(kmax - kmin + 1)^2} \tag{3.11}$$

The residue distribution allows us to establish our criterion based on the percentage of measurements belonging to uniform MM. We may use the Tchebycheff's inequality (Meyer, 1970) to determine a threshold which should leave the 90% of the measurements belonging to our linear model above it, with $\mu + 3\sigma$ value. Using the values from eq. (3.11), eq. (3.12) presents the obtained threshold value.

$$thres = 2 - \frac{4}{N} + 3\sqrt{\frac{4}{N} - \frac{8}{N^2}} \tag{3.12}$$

This threshold depends on the resolution of the segment, which also influences the residue value in (3.10). It is interesting to notice that the highest threshold value is reached with the lowest resolution. This is a logical result, since to be able to keep our percentage of uniform measurements correctly classified (usually called True Positives Rate or TPR), which has been fixed with the inequality at 90%, with short segments, we need to have a high threshold, in order to counteract the noise effects (while longer segments are more resistant to that noise and thus the threshold value may be lower).

It is necessary to determine how precisely the chosen χ^2 distribution represents the normalized BLUE residue in non-uniform trajectories with estimated covariance matrix. The following figures we compare the results obtained using equation (3.12) with the optimal result of the threshold choice (dotted lines), manually chosen to obtain the highest possible TPR while False Positives Rate (FPR, measurements not belonging to the uniform model

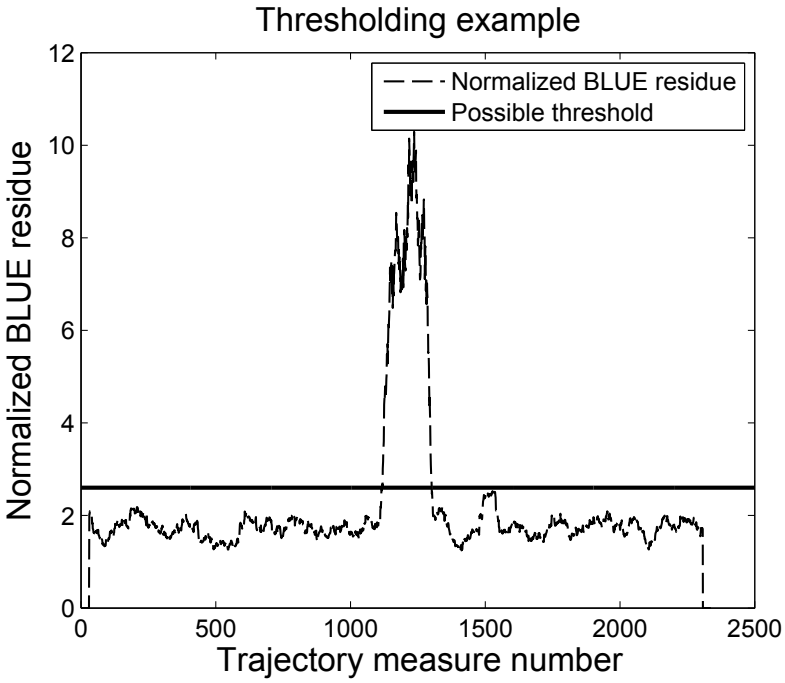


Figure 3.5: Threshold choosing example

misclassified) remains in a zero value. Figure 3.6 shows the used trajectories for this comparison, whereas figure 3.7 shows the actual comparison for the proposed trajectories between the optimal TPR and the one obtained with eq. (3.12) for increasing threshold values.

In the two trajectories in figure 3.7 we may appreciate two distortion effects introduced by the presented approximation. The turn trajectory shows an underestimation of the TPR value due to the inexactitude in the covariance matrix R_k. This inexactitude assumes a higher noise than the one which is present in the trajectory, and thus will require the choice of a higher threshold than necessary in order to obtain the desired TPR margin.

In the racetrack trajectory we perceive the same underestimation at the lower values of the threshold, but then the approximation result crosses the optimal results and reaches a value over it. This is caused by the second distortion effect, the maneuver's edge measurements. The measurements close to a maneuver beginning or end tend to have a higher residue value than the theoretical one for a uniform trajectory (due to their proximity to the non-uniform segments), making us increase the threshold value to classify them correctly (which causes the optimal result to show a lower TPR in the figure).

These two effects show that we may need a heuristic tuning in our χ^2 distribution in order to adapt it to these distortion effects. For our PLR approach, it is enough to higher the threshold value to $\mu+5\sigma$, knowing that we may misclassify some non-uniform measurements close to the points where the MM changes (considering that this does not have an important penalization impact).

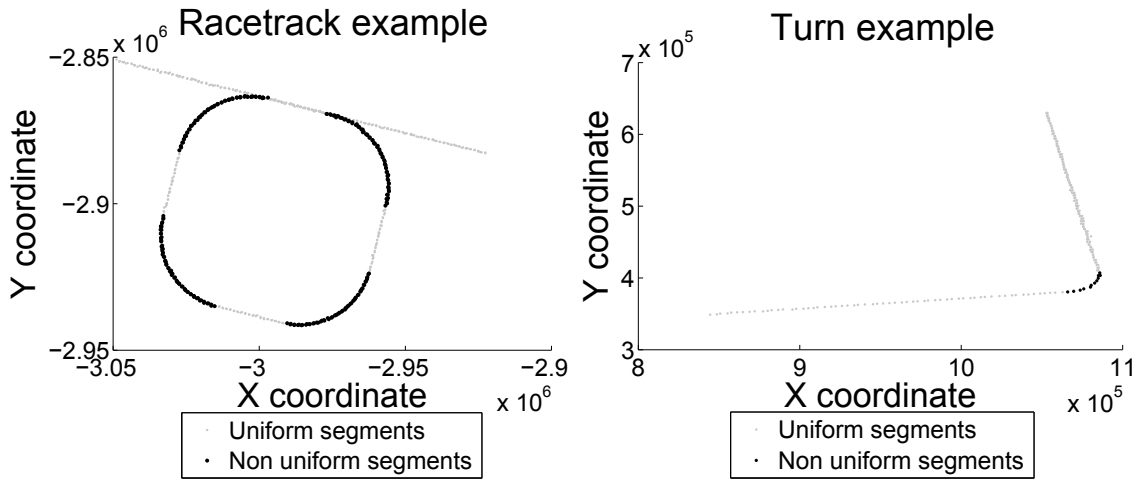


Figure 3.6: Considered trajectories for the threshold choice effects analysis

3.4.3 Algorithm definition

After the coverage of the theoretical considerations behind the segmentation technique proposal have been covered, the pseudocode for it will be specified in current section. The functions required for this pseudocode are programmed with the proposed equations in sections 3.4.1 and 3.4.2. Algorithm 4 presents the first phase of our technique.

The second phase of the algorithm applies the bottom-up technique (which is introduced in section 2.8.2 and will be more deeply analyzed in section 7.3, where a multiobjective version of the algorithm will be presented) and corrects the segmenting points obtained to their positions in the original trajectory, providing the final output of the algorithm as a series of segmenting points.

3.5 Computational complexity analysis

The complete complexity analysis for the three different traditional techniques can be found in (Keogh et al., 2003). The results presented are the following:

- Sliding-Window = $O(Ln)$
- Top-Down = $O(n^2K)$
- Bottom-Up = $O(Ln)$

where n is the number of measurements in the time series, K is the number of segments and L is the mean length of the obtained segments. It is important to notice that, for each complexity order, there is at least one parameter not known a-priori, (either K or L) which makes this complexity orders harder to be accurately established.

Algorithm 4 Hybrid Local Residue Analysis Algorithm, first phase

```
Input: time sequence (a_1, ..., a_k), time_length_window
Output: uniform segments (s_1, ..., s_k), non uniform segments (sn_1, ..., sn_m)
classifications = uniform_segments = non_uniform_segments = empty set
initial_point = 1
current_point = 1
sequence_length = length (time sequence)
while current_point <= sequence_length do
  current_segment = obtain_segment (time_sequence, current_point, time_length_window)

  current_length = length(current_segment)
  current_threshold = obtain_threshold (current_length)
  current_residue = obtain_residue(current_segment)
  if current_residue > current_threshold then
    add(classifications, non_uniform_class)
  else
    add(classifications, uniform_class)
  end if
  if current_point > 1 && (classifications (current_point) != classifications (cur-
  rent_point-1) || current_point == sequence_length) then
    if current_point == sequence_length then
      if classifications(current_point) == uniform_class then
        add (uniform_segments, current_point)
      else
        add (non_uniform_segments, current_point)
      end if
    else
      if classifications(current_point) == uniform_class then
        add (non_uniform_segments, current_point)
      else
        add (uniform_segments, current_point)
      end if
      initial_point = current_point-1
    end if
  end if
  current_point++
end while
```

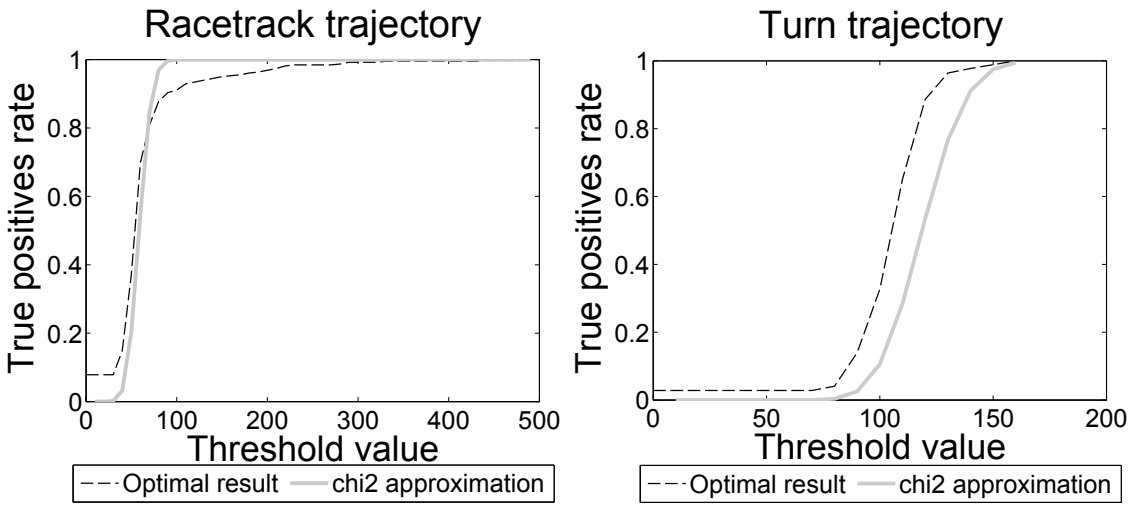


Figure 3.7: χ^2 approximation comparison.

3.5.1 Hybrid Local Residue Analysis Algorithm

The proposed algorithm involves two different phases, each of them based on different approaches. We will present them separately in order to obtain the complexity order. There are, as well, two different main steps involved in the first phase of the algorithm:

- *Threshold value:* for a fixed window size, this parameter can be computed only once, with a constant order complexity. If the window is not fixed (or established with time boundaries, which result in windows with different sizes at every measurement) this involves computing the threshold n times, where n is the number of measurements, thus adding an $O(n)$ term.

- *Residue value obtaining:* The residue has to be calculated, with a certain window size, at every value of the time series. Calculating each residue involves a cost of $O(w_l)$, where w_l represents the *window_length* involved, and applied to each value of the time series, involves a cost of $O(w_l n)$.

The second phase shows the computational complexity of the bottom up algorithm, which is, as presented in the previous section, $O(Pq^2)$, applied t times, where t is the number of non-uniform segments in the trajectory, P is the mean length of the sub-segments in those segments and q their number of measurements, giving us a complexity order for this second phase of $O(tPq^2)$.

The final complexity order of the trajectory is, adding the terms from the previous two phases, $O(w_l n)+O(tPq^2)$. Considering that P is a small value (as the secondary segmentation is applied to non-uniform sections of the time series, which cannot be well approximated by long uniform segments), $q <<< n$ and $t <<< w_l$ (with the possible exception of extremely long time series, where the value of n compensates for the possible increase in the value of t), we can determine that the dominating order term is $O(w_l n)$, being this the complexity of our proposed approach.

Compared to the presented complexities of the traditional techniques, the proposed solution shows the advantage of having parameters that can be either fixed by the user or accurately approximated, opposed to some of the terms presented by traditional techniques not known *a priori*, such as the mean length of the final segments or the number of these segments.

3.6 Experiments

3.6.1 Quality measurements and algorithm configuration

In previous sections of this work (particularly during the analysis of segmentation techniques and its formalization in section 3.3) the importance of the number of segments has been repeatedly stated, along with the absence of its value which is usually found in available references. But, for current domain application, there are additional considerations which must be included in the quality indicators to perform an accurate comparison.

Evidently, as was introduced, the objective of any segmentation process is, usually, reducing the amount of information while keeping a record as similar as possible, compared to the original data (even though additional processes with different objectives can be performed over the transformed data). This means that, in a noisy domain such as the one presented, we would like to reduce the effect of the noise as much as possible (whenever it can be differenced from the actual data of our aircraft). Considering the MM's presented, if we divide uniform segments into several different sub-segments, that division is performed due to the noise position changes, and thus, we are including additional segments which are a waste of data (along with the additional problems for any processing which might be performed afterwards).

That fact can lead to misleading values in the *total_error* metric, which contains the deviation of the regression line with respect to the noisy samples. Oversegmentation would reduce the residual, with an evident effect of over-fitting to the noise contained in the series. Figure 3.8 presents an example of a uniform time series to which Gaussian noise has been added, with $\mu = 0, \sigma^2 = 1$ along with its ideal segmentation (based on the original time series previous to the noise addition) and a segmentation result based on interpolation using segments a fixed length of three measurements.

The ideal segmentation in figure 3.8 identifies correctly the time series as a single segment, presenting a *total_error* value of 14.91, while the interpolation segmentation, which oversegmentates the trajectory into different segments, presents a *total_error* value of 10.69. This simple example shows that the quality of a segmentation on noisy time series should not be measured by means of a *total_error* metric (at least over uniform segments). According to this, we will introduce two different metrics, one related to the segmentation quality over non-uniform segments and a different one for those who were performing a uniform MM in the original trajectory.

For non-uniform segments, we will include the *total_non_uniform_error*, which is the *total_error* metric but only applied to those measurements where the aircraft was performing a non-uniform MM (lacking a better quality metric for those non-uniform segments). This metric assesses the behavior of segmentation algorithm under a situation in which the series should be divided to avoid the deviation produced by a linear model in situations in which it is not applicable.

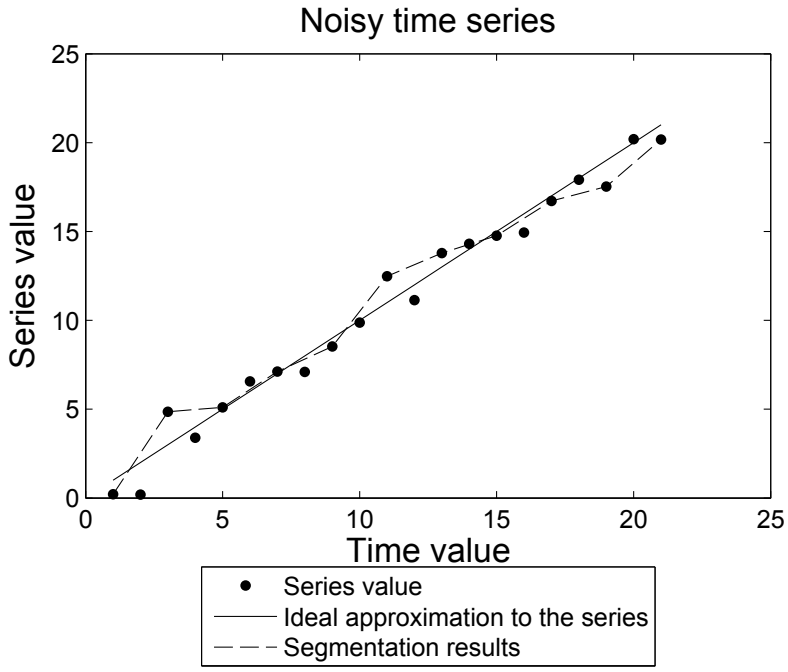


Figure 3.8: Two example segmentations for a completely uniform noisy time series.

On the other hand, for the quality assessment of uniform segments, we will introduce the Uniform Segmentation Ratio (*USR*), which, dividing the ideal number of uniform segments (those performed by the aircraft) by the number of segments obtained by the technique, tries to measure the level of over-segmentation obtained during uniform MMs. This quality metric is defined in (3.13).

$$USR = \frac{\text{number of original uniform segments}}{\text{resultant segments during uniform MMs}} \tag{3.13}$$

The ideal value for this indicator is 1. Lower values indicate oversegmentation on those uniform MM's, while a higher value is only possible if segments exhibiting a non-uniform MM are approximated in a single segment, with a severe increase in the approximation error. The segments taken into account to be computed in the previous ratio are those which have any measurement recorded while the aircraft was performing a uniform MM. It is interesting to realize the complementary nature of the two previous figures of merit. An algorithm prone to oversegmentation will have a very low uniform segmentation ratio and, conversely, an algorithm prone to keep long segments through the whole series will tend to obtain an unfeasible error value during maneuver sections of the time series.

The final value included for this comparison is the *running_time* of the technique. Obviously, the actual value of this metric depends on different factors, such as the programming language chosen, and cannot be interpreted as an absolute value, but it can be used as a comparative value between different techniques or different configurations over the same technique.

Four different quality metrics will be used to measure the performance of a given algorithm: *total_non_uniform_error* (accuracy in the representation of those segments which lack a uniform model), *number of segments* (overall cost of the segmentation results), *USR* (ac-

curacy during uniform segments), and *running time* (computational cost of the segmentation results, an indicator of the feasibility of the application to long time series).

There are configuration issues to be faced which will involve the quality of the results. During the presentation of the traditional techniques, the only shared configuration value was *max_segment_error* (sliding window cannot provide a *total_error* boundary due to its online nature), so we will choose the value for this parameter for the three techniques. An approach could be to determine different values for the different trajectories, in order to optimize the performance of the algorithms in each particular case. This might be achieved by means of the domain knowledge, for simulated trajectories, of the typical durations of maneuvers performed by aircrafts following different trajectories.

Such an approach would require for each trajectory and algorithm with its own configuration values, meaning that they would be inapplicable to real trajectories after this initial definition phase (where we would have no *a-priori* knowledge). This instance based configuration issue is similar to those faced on the stopping criteria presented in chapters 4 and 5, which had to overcome the traditional criterion of stopping according to a measure of proximity to an *a-priori* known optimal solution (usually named reference or global criteria). To prevent this, a single *max_segment_error* value will be determined to be applied to any trajectory or algorithm, in order to test their performance as a whole.

Once that decision has been taken, the choice of that threshold is not trivial either. We have different techniques, different trajectories and, most importantly, different metrics which have to be optimized jointly. There is also an additional consideration. Choosing a fixed *max_segment_error* tends to set a threshold on the maximum length of the obtained segments (considering that every measurement of the time series carries an error due to the noise presence), leading the algorithm to obtain shorter segments. To prevent this behavior, we will use the *max_mean_segment_error* value instead. The idea for this parameter is to allow segments to be as long as possible, by setting a threshold over the mean value of the different errors of the measurements belonging to a segment, eliminating the implicit length boundary which *max_segment_error* exhibits. According to this, the three traditional segmentation techniques will be provided with only one parameter, the *max_mean_segment_error*, and different values will be tested regarding this parameter, in order to compare their applicability.

For the proposed technique, the time length of the window (according to equation 3.5) has to be set. As introduced in the domain presentation (section 3.2), this value is chosen based on the non-uniform MM characteristics. The value chosen for the data set proposed is 60 seconds. The bottom-up technique also requires a *max_mean_segment_error* value which, in this case, it is set to 300 meters.

Finally, statistical tests are required to determine the quality of the different compared techniques. This introduces the difficulty of quantitatively determining the quality of different multi-objective solutions and their comparison for quality assessment purposes (Zitzler et al., 2008). Basically, this issue can be approached by the use of a quality indicator, which can reduce the different objectives to a single value and performing a statistical test to determine whether the different result sets can be considered to belong to the same distribution. As seen in section 2.6, this is not the only way to perform such a comparison (for instance, attainment functions and their related tests were covered in section 2.6.2), but quality indicators is the most extended among them.

Quality indicators were designed for the comparison of different Pareto Fronts, but

current domain only requires one solution for each trajectory in the data set, which simplifies the difficulties of the comparison. Their related theory has been covered in section 2.6.1, and specific instances of them will be presented in sections 5.2.1 (hypervolume) and 5.2.1 (epsilon indicator, both in multiplicative and additive terms). Also, we will use three objective functions (*USR*, *number of segments* and *total_non_uniform_error* quality measurements). Considering these simplifications we will stick to a unary hypervolume quality indicator (Zitzler & Thiele, 1998) for the individual estimation of the quality of the obtained solutions (for a more thorough analysis of this indicator, along with its binary formulation, the reader may see section 5.2.1. The general formulation of this indicator is included in equation 3.14.

$$I_h(A) = volume\left(\bigcup_{\forall z \in A; \forall y \in N} hypercube(a, n)\right) \tag{3.14}$$

This estimator requires the choice of a *nadir* point, which is the worst possible solution for the problem. The choice of these points is itself an issue (Ehrgott & Tenfelde-Podehl, 2003). The *total_non_uniform_error* value of our chosen nadir points will be a theoretical maximum error obtained by joining the first and last points of the current time series with a segment and calculating the error of the different points in the time series as the distance to that segment. The highest number of segments considered will be the number of points in the time series minus one (representing the worst oversegmentation situation possible, where a segment joins every pair of adjacent points).

The *USR* value for the nadir points is a little harder to obtain since we may degrade its value oversegmenting segments with uniform MM or introducing into them segments with a non-uniform MM. Considering only the oversegmentation, the nadir point value for its *USR* component would be zero, but there is not such a boundary for the possible values of this indicator considering the possible errors in the segmentation of non-uniform MMs. In the results for the dataset presented (tables 3.1-3.3) the worst possible result obtained regarding this error source is 2, so these values will be converted to the [0, 1] interval considering a worst value of 2.01 (Eq. 3.15). This means that a *USR* value of 2 is treated in a similar way in the results as an oversegmentation value of 0.01, and the nadir point value for *USR* is 0. In order to normalize the hypervolume values, the *total_non_uniform_error* and *number of segments* values for the different techniques will be normalize according to the worst possible results presented, so that the nadir point values for both of them will be 1 (Eq. 3.16).

$$normalized\ USR = \begin{cases} USR & \text{if } USR \leq 1 \\ 2.01 - USR & \text{if } USR > 1 \end{cases} \tag{3.15}$$

$$nadir\ point \rightarrow \begin{cases} USR = 0 \\ t\_n\_u\_e = 1 \\ number\ of\ segments = 1 \end{cases} \tag{3.16}$$

Considering the normalized values presented, the hypervolume indicator (which, in this case, is a three dimensional volume) can be calculated with equation (3.17). Over the hypervolume values for the dataset, the Wilcoxon test (Hollander & Wolfe, 1999) test will be applied to determine their statistical significance.

$$\begin{aligned} hypervolume =& (t\_n\_u\_e_{nadir} - norm.\ t\_n\_u\_e) * (norm.\ USR - USR_{nadir}) \\ & * (n\_o\_s_{nadir} - norm.\ n\_o\_s_{)} \end{aligned} \tag{3.17}$$

3.6.2 Data set definition

The data set used is based on eight trajectories covering the different MM described in the segmentation issues section. The complete dataset used is shown in figure 3.9.

These simulations cover the casuistry of the domain, with the specified characteristics presented, and allow us to determine the validity of the different included techniques. For the computation of the proposed ratio, completely uniform trajectories (trajectories 3 and 4) show a difficulty, as all their measurements belong to a uniform MM, so that the *total_non_uniform_error* value, regardless of the segmentation performed, will always be 0.

3.6.3 Traditional techniques results

Tables 3.1 and 3.2 show the results of the presented classical segmentation techniques (introduced in section 2.8.2) applied to the proposed data set. The tray column shows the trajectory number (according to figure 3.9), *m.m.e.s* stands for *max_mean_segment_error*.

There are some interesting observations to be made regarding the results exposed in tables 3.1 and 3.2. First of all, the effect of the *max_mean_segment_error* is opposite in the two introduced quality indicators: choosing higher values allows the technique to improve the segmentation results in the uniform segments (reflected in the USR values) but introduces poorer results in the segmentation of *non_uniform_segments*. This introduces irresolvable configuration issues, due to the lack of mechanisms in these techniques to differentiate uniform and non_uniform segments.

Regarding the previous configuration issue, it is also noticeable that these algorithms are not able to correctly segmentate the uniform trajectories (trajectories 3 and 4) with any of the tested *max_mean_segment_error* values. The highest value for this parameter, 800, lead to a segmentation of completely uniform trajectories into two segments (probably an acceptable result) but made the techniques obtain very inaccurate results in accelerated trajectories (7 and 8), obtaining only one final segment in them.

Offline algorithms, reported to be the most accurate ones due to their global knowledge of the time series, reach inadmissible running time levels in some trajectories when faced with values which affect greatly their complexity. The Top Down algorithm has difficulties dealing with low *max_mean_segment_error* values (trajectories 4 and 6 with MMSE=200), while the bottom up technique increases its running time noticeably in the presence of a large number of measurements (trajectories 4, 6, 7 or 8).

These high running times may make them inapplicable to long real trajectories. Also, there are additional issues related to the extremely high recursion level that the Top Down algorithm has to reach in order to perform its segmentation in trajectories with a high number of measurements, which may lead to the algorithm malfunction. These issues may require the development of alternative, non-recursive implementations (if possible) to guarantee the feasible application of this algorithm.

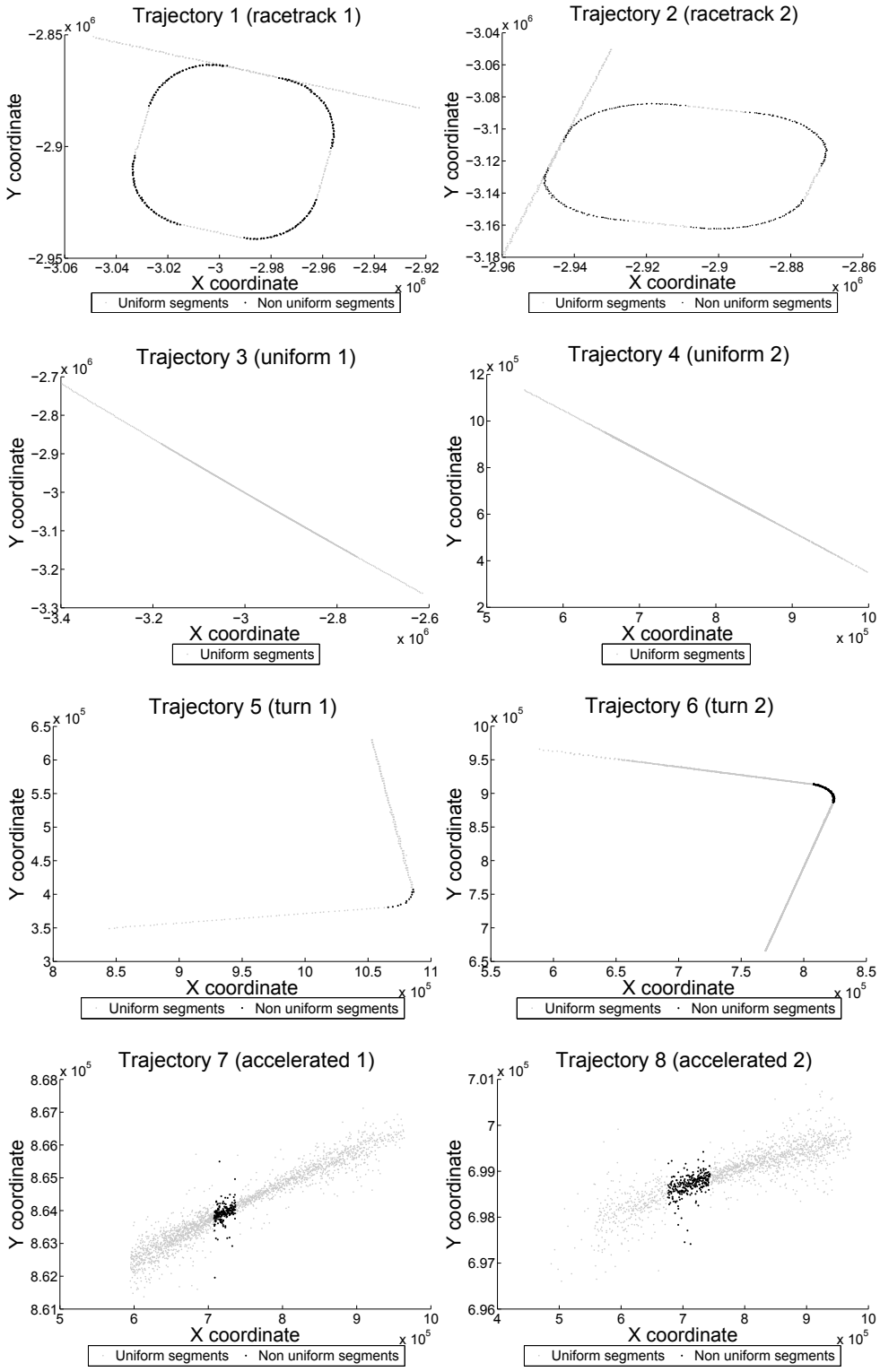


Figure 3.9: ATC trajectory dataset used for evaluation purposes

Table 3.1: Sliding window and Bottom-up segmentation techniques results for different *max_mean_segment_error* values

Tray	m.m.s.e.	Sliding Window				Bottom Up			
		n.u.error	n. segm	USR	Run. time	n.u.error	n. segm	USR	Run. time
1	200	55680	93	0.12	0.06	42019	92	0.13	15.68
1	500	152973	27	0.45	0.06	109954	24	0.63	16.20
1	800	263000	13	1.00	0.06	184770	13	0.83	16.10
2	200	55650	255	0.03	0.07	46422	246	0.03	14.47
2	500	158135	58	0.12	0.06	93688	57	0.13	16.68
2	800	257286	20	0.42	0.06	197702	16	0.71	16.85
3	200	0	168	0.01	0.08	0	114	0.01	27.08
3	500	0	15	0.07	0.09	0	2	0.50	27.54
3	800	0	2	0.50	0.09	0	2	0.50	27.52
4	200	0	556	0.00	0.21	0	539	0.00	149.83
4	500	0	54	0.02	0.25	0	36	0.03	159.56
4	800	0	3	0.33	0.32	0	1	1.00	160.77
5	200	1995	148	0.01	0.03	2000	138	0.02	1.70
5	500	9487	114	0.02	0.03	8834	110	0.02	1.96
5	800	23122	93	0.02	0.03	11877	93	0.02	2.09
6	200	54463	328	0.01	0.25	34698	288	0.01	284.44
6	500	203155	19	0.14	0.34	182421	5	1.00	286.63
6	800	335674	4	1.00	0.37	445360	3	1.00	286.33
7	200	32381	49	0.04	0.29	32718	38	0.05	179.64
7	500	33512	1	2.00	0.35	33512	1	2.00	180.18
7	800	33512	1	2.00	0.36	33512	1	2.00	180.53
8	200	38162	62	0.03	0.18	39467	1	2.00	103.50
8	500	39467	1	2.00	0.23	39467	1	2.00	103.54
8	800	39467	1	2.00	0.23	39467	1	2.00	103.42

3.6.4 Hybrid Local Residue Analysis Segmentation results

Table 3.3 presents the results obtained by the proposed algorithm. The main handicaps which were detected in the results presentation of traditional techniques have been properly corrected: those trajectories which were originally completely uniform are now correctly segmentated into a single segment (trajectories 3 and 4, where traditional techniques showed a minimum number of 2 segments), accelerated trajectories are detected to include non-uniform segments and segmentated accordingly (where certain configurations of the bottom-up and sliding window algorithms obtained a *USR* value of 2, bypassing the accelerated MM) and the running time remains at an allowable maximum value (18.57 seconds, while the bottom-up algorithm showed a maximum value of 286.33 seconds and the top-down approach a maximum value of 162.41). It is also noticeable that the trade-off among the different values of the metrics (even though its configuration parameters are fixed, while traditional techniques have been tested with a different set of values for their configuration) is consistently better than the one present in traditional techniques.

3.6.5 Results comparison

Even though the individual analysis of the results has already been presented in sections 3.6.3 and 3.6.4, along with other associated tables 3.1-3.3, it is necessary to compare some of the quality indicators results for the different techniques graphically, in order to complete this results presentation section. The objective of this approach is to present a general analysis of the performance achieved by the different techniques to support the choice made for the most promising alternative, followed by the complete comparison versus our presented technique in order to validate its results.

This graphical overview will present firstly the comparison of the different techniques for a concrete trajectory (one of the racetracks, trajectory 2). In figures 3.10 and 3.11 it can be observed that the proposed technique achieves much better results according to the quality metrics, especially regarding (as was commented after the presentation of the results tables) the trade-off in their different values. To obtain a better result in terms of *total_non_uniform_error* or *uniform segmentation ratio* traditional techniques must degrade the value of the complementary evaluated metric, obtaining unfeasible solutions. Using intermediate configurations (parameter MMSE set to 500), the proposed technique obtains better solutions for the two metrics, also obtaining a smaller number of segments as its output. Among the traditional techniques, bottom up segmentation seems to achieve the best results, so we will choose it with a MMSE value of 500 as the most promising technique.

Once the most promising traditional technique has been chosen, the next comparison step is the presentation of the results for the whole set of trajectories comparing the bottom-up technique with the indicated configuration with HLRA's results. These comparisons are shown in figures 3.12 and 3.13. The results seem to be conclusive: in all the different trajectories presented, the proposed technique achieves better results that the bottom up algorithm, being specially remarkable in some cases, such as the uniform trajectories (HLRA-T3, HLRA-T4, where the degree of oversegmentation with bottom up technique is very high, while HLRA detects correctly a single segment) or the turn ones (where the bottom up technique presents extreme values in either the number of segments, HLRA-T5, or the *total_non_uniform_error* value, HLRA-T6). It is also interesting to highlight that the results of the proposed technique are satisfactory for all the different trajectories, being suitable for

Table 3.2: Top down segmentation technique results for different max_mean_segment_error values

Tray	m.m.s.e.	Top Down			
		n.u.error	n. segm	USR	Run. time
1	200	57788	94	0.12	0.90
1	500	94136	38	0.29	0.77
1	800	151262	15	0.71	0.63
2	200	65193	199	0.04	3.23
2	500	112310	116	0.07	2.86
2	800	160489	43	0.18	2.58
3	200	0	223	0.00	4.97
3	500	0	2	0.50	0.19
3	800	0	2	0.50	0.19
4	200	0	786	0.00	59.08
4	500	0	5	0.20	2.60
4	800	0	2	0.50	0.65
5	200	9419	87	0.02	1.09
5	500	9419	87	0.02	1.09
5	800	26722	78	0.03	1.07
6	200	27816	769	0.00	162.41
6	500	209334	4	1.00	1.83
6	800	209334	4	1.00	1.83
7	200	29359	294	0.01	20.13
7	500	33504	2	1.00	0.71
7	800	33504	2	1.00	0.71
8	200	39467	2	1.00	0.46
8	500	39467	2	1.00	0.46
8	800	39467	2	1.00	0.46

Table 3.3: HLRA segmentation technique results for the complete dataset.

Tray	HLRA algorithm			
	n.u.error	n.segm	U.S.R.	Run. time
1	58979	47	0.29	2.20
2	70874	51	0,45	2,27
3	0	1	1,00	1,34
4	0	1	1,00	8,32
5	23471	8	0,67	0,18
6	52488	23	0,25	18,57
7	32282	7	1,00	10,24
8	36748	5	1,00	7,69

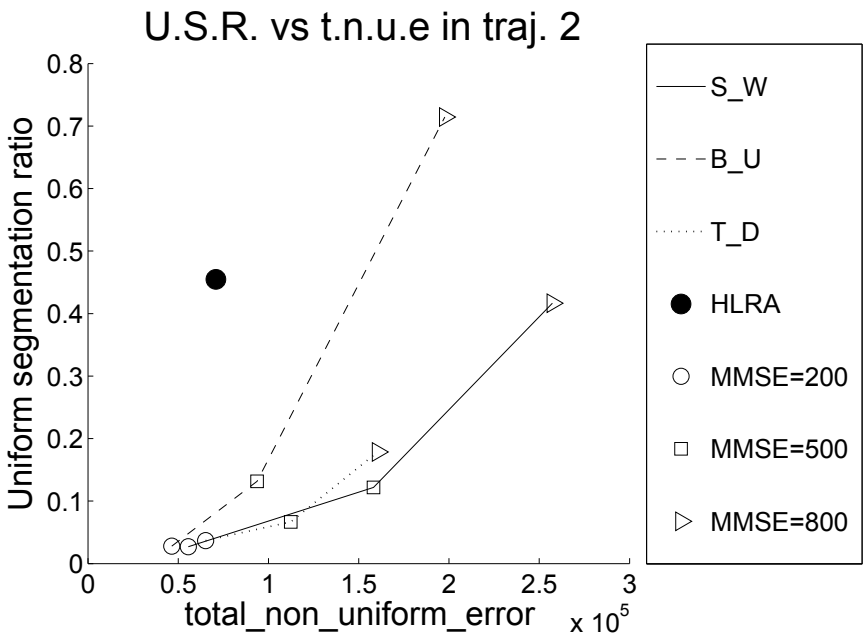


Figure 3.10: *uniform segmentation ratio* and *total_non_uniform_error* values comparison in trajectory 2

any of them.

Finally, the statistical significance of the results must be proved. To do so, as explained in section 3.6.1, we will calculate the hypervolume indicator values for the results over the different trajectories in the data set for the already chosen most promising technique and HLRA, in order to apply the Wilcoxon test to state whether the result improvements of HLRA are significant or not. The results of the normalized values for the chosen quality measurements, along with their associated hypervolumes are presented in tables 3.4 and 3.5.

Running the Wilcoxon test over the hypervolume values in tables 3.4 and 3.5, the p-value obtained is 0.0368. This means that with the usual 5% significance level, the null hypothesis that both datasets came from the same distribution can be rejected. In fact, the significance level can be lowered down to a 4% value and still reject the null hypothesis. This result implies that the improvements are statistically significant.

3.7 Conclusions

This chapter has introduced the difficulties faced by time series segmentation algorithms on domains with long time series exhibiting noisy measurements. Noise degrades the segmentation performance over uniform sections of the time series (which should be packed into a single segment), while the large number of measurements prevents the application of techniques based on global approaches (due to the running time or the recursion level required). These difficulties are faced with the proposed Hybrid Local Residue Analysis technique, based on two phases: the first one differentiates uniform and non-uniform segments in the trajectory, while the second one segmentates the identified non-uniform segments one by one. Noise

Table 3.4: Normalized quality measures and associated hypervolume values for bottom up technique with mmse=500

Trajectory			Bottom Up mmse=500			
id	max error	max segm	norm t n u e	norm n o s	norm USR	Hypervolume
1	9,05E+11	651	1,22E-07	0,036866359	0,625	0,601958452
2	1,77E+13	665	5,30E-09	0,085714286	0,131579	0,120300799
3	3,27E+10	851	0	0,002350176	0,5	0,498824912
4	6,38E+09	2056	0	0,017509728	0,0277778	0,027291418
5	6,17E+12	252	1,43E-09	0,436507937	0,0192308	0,010836403
6	1,15E+14	2747	1,58E-09	0,001820167	1	0,998179831
7	3,26E+08	2178	1,03E-04	0,000459137	0,01	0,009994382
8	3,49E+08	1650	1,13E-04	0,000606061	0,01	0,009992808

Table 3.5: Normalized quality measures and associated hypervolume values for HLRA technique

Trajectory			HLRA			
id	max error	max segm	norm t n u e	norm n o s	norm USR	Hypervolume
1	9,05E+11	651	6,52E-08	0,072196621	0,294118	0,272883657
2	1,77E+13	665	4,01E-09	0,076691729	0,454545	0,419685156
3	3,27E+10	851	0	0,001175088	1	0,998824912
4	6,38E+09	2056	0	0,000486381	1	0,999513619
5	6,17E+12	252	3,80E-09	0,031746032	0,666667	0,645502966
6	1,15E+14	2747	4,55E-10	0,00837277	0,25	0,247906807
7	3,26E+08	2178	9,90E-05	0,003213958	1	0,996687373
8	3,49E+08	1650	1,05E-04	0,003030303	1	0,996864615

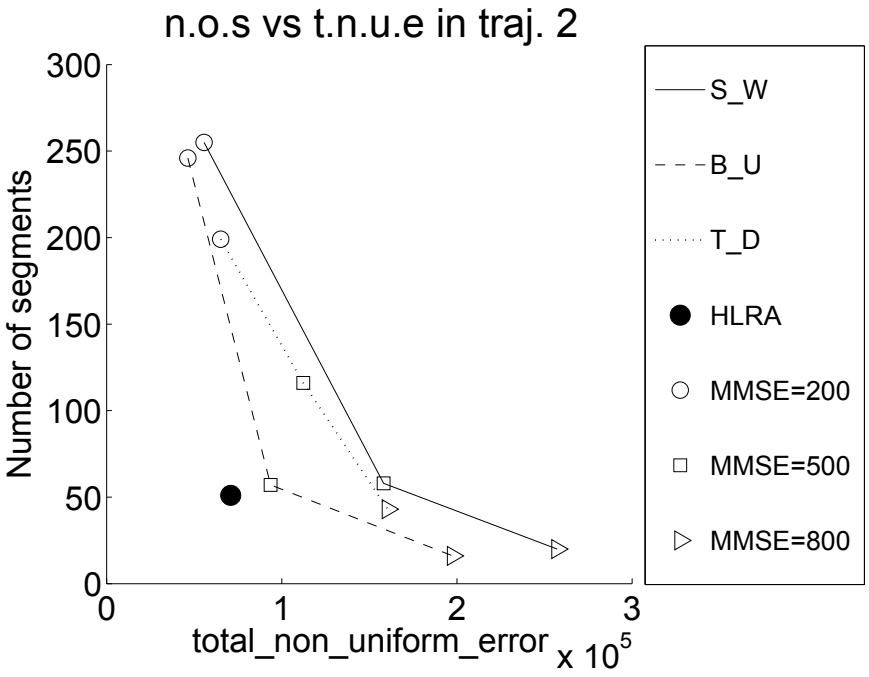


Figure 3.11: *number of segments* and *total_non_uniform_error* values comparison in trajectory 2

information is introduced in the initial separation into uniform and non-uniform segments, and the individual approach to each non-uniform segment separately allows the techniques to deal with time series which were not approachable without this pre-segmentation (due to the huge decrease in the number of measurements in each of those individual non-uniform segments).

The problem formulation has resorted to a multi-objective, constrained approach, requiring as well a modification of traditional error indicators, performed in order to deal with the noise in pure traditional techniques (basically establishing a threshold over the max error in mean over the window, instead of absolute values), and performance metrics are introduced in order to measure the quality of the different compared techniques. These performance metrics have to determine not only the compromise present between cost and performance metrics, but also the use where the measured cost is spent (whether the segments are being used to cover only the noise perturbations or in sections of the time series where really a higher number of segments is required).

The results obtained with the Air Traffic Control domain dataset show that the HLRA technique can take advantage of the noise information in order to perform the initial division accurately and afterwards apply bottom up segmentation to obtain a fine segmentation over the non-uniform sections, providing considerably better results that traditional techniques for the different quality indicators presented.

Along with the quantitative objectives of the work, represented by the segmentation results already commented, this chapter the application of artificial intelligence tools to improve the heuristic guided pattern recognition issue which is at the core of segmentation problems, along with the use of quality metrics obtained from the multi-objective evolutionary

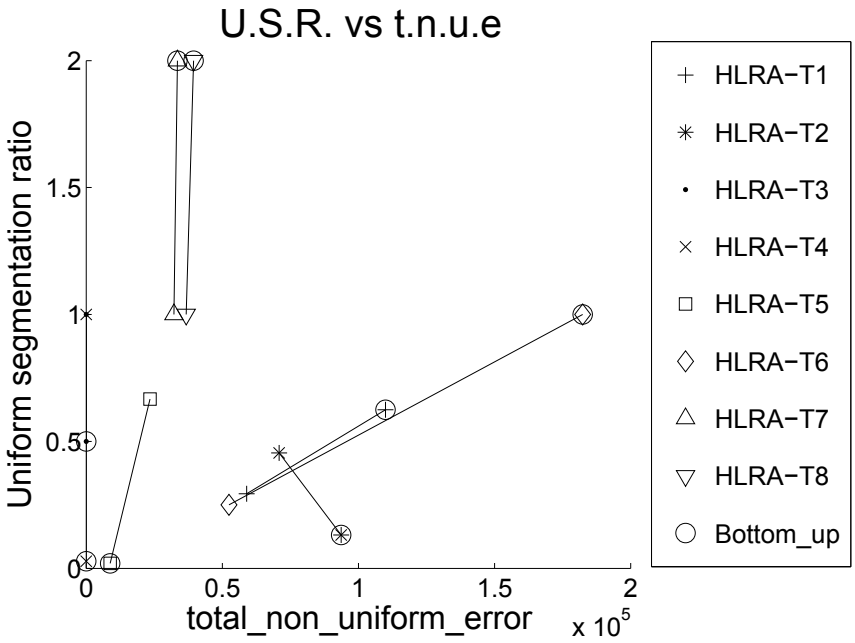


Figure 3.12: *uniform segmentation ratio* and *total_non_uniform_error* values comparison for the bottom-up technique with MMSE=500 and the proposed technique applied to the whole data-set

algorithms domain to determine statistical significance of the results (a matter thoroughly analyzed in section 3.6.1). Also, the results obtained may be used as a data source to improve the performance of reconstruction approaches in the air traffic control domain, and, at the same time, noise handling techniques are introduced to the general PLR segmentation issue.

Concerning the overall objectives of this thesis, this chapter has established the difficulties faced by specific heuristic methods applied to a segmentation domain and the effort required to propose novel heuristics according to these domains. Also, it provides a key insight into the process itself, which can be used for instance, at the definition of initialization methods for the evolutionary proposal (which will be introduced in section 6.4.2). Finally, the required multi-objective quality indicators point to the multi-objective nature of the segmentation problem.

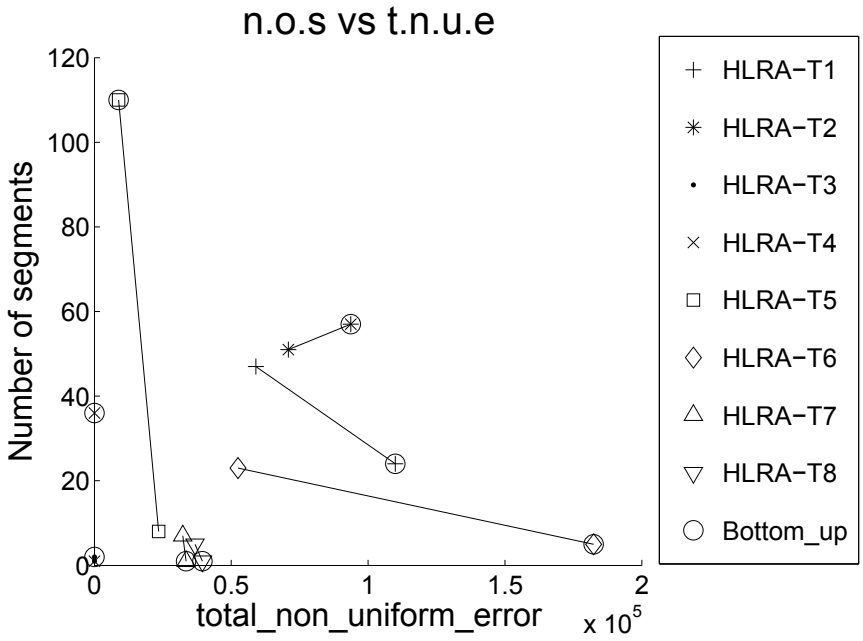


Figure 3.13: *number of segments* and *total_non_uniform_error* for the bottom-up technique with MMSE=500 and the proposed technique applied to the whole data-set

4

Single-objective stopping criteria for evolutionary algorithms

> *Because that was then and this is now Because the past is gone , even though it defines the present*

Stephen King, *Doctor Sleep*, 2013

This chapter studies single-objective stopping criteria, detailing the different tools and approaches required for this issue, with a special focus on diversity management and the pasive or active role of the criterion. More specifically, the chapter will be based on the modification of a memetic algorithm: ESLAT, Evolutionary Strategies Learned with Automated Termination Criterion (Hedar & Fukushima, 2006) to increase its robustness and improve the presented stopping criterion to implement the R-ESLAT algorithm (section 4.2) and finally a novel stopping criterion for single objective evolutionary strategies will be presented (section 4.3). This criterion will attempt to control the stagnation situation at the same time that it measures it, attempting to prevent early convergence through the population's diversity control. The main references for this chapter are (Guerrero et al., 2014a, 2011b, 2012c).

4.1 Introduction

Evolutionary algorithms, as covered in chapter 2, are probably one of the most versatile and used tools in artificial intelligence to deal with optimization problems, as shown in recent compilations of applications (such as (Affenzeller & Winkler, 2009)) and the exponential growth of their application to different practical domains, ranging from bankruptcy prediction modeling (Shin & Lee, 2002) to general classifier systems learning (Lanzi, 2009). The increase in the computational resources of computers and the increasing number of parallel implementations (Cantu-Paz, 2000) have lead this growth, making them more appealing for practitioners focused on solving particular problems, rather than theoretical research of the algorithms themselves. There are, however, a number of issues which are still a drawback for these applications.

Local optima constitute a drawback for evolutionary algorithms, since they do not provide (as most metaheuristics (Talbi, 2009)) a measurement of the proximity of the solutions found to global optima, providing a best-effort approach. Early convergence arises as a concern

regarding this topic, being closely related to the diversity preservation in the populations as the evolution process advances. Many approaches have been proposed to deal with this issue, from the restriction of certain operator applications (such as the incest prevention proposed in (Eshelman & Schaffer, 1991)) or multi-objective approaches (Coello et al., 2007) where the diversity of the population is treated as an additional objective function (Toffolo & Benini, 2003).

Any optimization algorithm has to deal with two complementary processes: exploration and exploitation. Evolutionary algorithms (and, in general, population based approaches) are considered to provide good results for the exploration component of the optimization process, whereas their results in exploitation are not so consolidated. This topic was covered in section 2.3, where the capabilities of the different families of metaheuristics were detailed. Particularly, figure 2.5 shows a graphical overview of the different strengths of these different metaheuristic families regarding exploration and exploitation, named diversification and intensification respectively.

Local search techniques (Hoos & Stützle, 2005) have complementary characteristics to EAs, excelling in the exploitation process but obtaining, in general, a poorer performance regarding their exploration capabilities. Memetic algorithms (Krasnogor & Smith, 2005), combine these two processes under a general cultural evolution framework. The combination of the two sets of techniques theoretically allows them to apply the exploitation capabilities of local search techniques to a better coverage of the search space.

General stopping criteria are also a concern for practitioners using evolutionary techniques. A general stopping process has to handle information provided by the previously stated processes, and determine whether the evolutionary optimization process will likely obtain better solutions according to them or not. In fact, this concern is shared by many iterative processes (Arioli et al., 1992), but the stochastic nature of evolutionary computation makes it probably more important and, at the same time, harder to solve. An overview of this topic, along with a classification of different approaches, has been presented in section 2.4.7. Traditional approaches to this issue set an *a-priory* budget of affordable computational cost (which may be expressed in terms of generations, function evaluations or even time) and stop the process once the budget has been fulfilled (what was defined in the presented classification as static, exhaustion based criteria). However, establishing this budget can be a difficult process for real problems. This concern is shared by single and multiobjective evolutionary algorithms. Some available approaches related to this multi-objective approach to stopping criteria were covered in section 2.7, and will be analyzed in chapter 5.

These previous concepts will be combined for the initial proposal of this chapter, the Robust Evolutionary Strategy Learned with Automatic Termination Criteria (R-ESLAT), a memetic algorithm with self-stopping capabilities, which performs a control over the population diversity and search space exploration. The proposal is based on the original ESLAT algorithm (Hedar & Fukushima, 2006), which introduced an evolutionary strategy along with local search procedures based on Matlab's *fminunc* function implementation and the simplex Nelder-Mead optimization method (Nelder & Mead, 1965) according to the modifications proposed in (Kelley, 2000). Additionally, it introduced the concepts of the *gene matrix*, which was used as a measure for exploration measurement and diversity control, as well as a mutation operator named *mutagenesis* which modified some of the values in the chromosome to cover specific zones of the search space. However, this technique did not specify some of the techniques required for its application, exhibited some difficulties (such as solutions

outside the search space) and its stopping criterion was focused only on the coverage of the search space.

The purpose of the R-ESLAT technique proposal included in this chapter is to analyze original the ESLAT algorithm, specifying and modifying the algorithm according to the exploration and exploitation capabilities required in order to improve its robustness and results quality, and finally compare the obtained results to one of the most extended and successfully applied evolutionary strategies: Covariance Matrix Adaptation (CMAES) (Hansen et al., 2003).

This initial proposal sets the basis for the final stopping criterion approach. The stopping criterion is extracted from R-ESLAT, and its passive role modified to an active one, in order to provide a diversity enhancer technique based on the mutagenesis mutation operator. This technique is based on stopping prevention concepts, and the required analysis of the stopping criterion associated to the presented approach. The presented technique will be inserted in a canonical evolutionary strategy, comparing the results of this strategy with and without the introduced artifact, in order to test the performance of the technique.

4.2 A robust memetic algorithm with self stopping capabilities: R-ESLAT

4.2.1 The original ESLAT algorithm

The ESLAT algorithm was originally proposed as a memetic algorithm that could overcome the slow convergence towards the minimum which canonical evolutionary strategies exhibit, controlling the achieved coverage of the search space and introducing a self-stopping criterion. The two introduced control artifacts for this purposes where the *gene matrix* (GM) and the *mutagenesis* operator.

Gene matrix and mutagenesis

The *gene matrix* is responsible of tracking the exploration process and keeping the diversity in the population. It is composed of n by m elements, where n is the number of genes in the chromosome and m is the number of sub-ranges in which the search space of that chromosome is divided. This matrix is initialized with zeros, and those zeros are updated to ones as elements with genes covering the different sub-ranges are found in the different populations as the evolution progresses. Figure 4.1 shows an example of a GM with two variables.

The *GM* is used, therefore, as a measurement of the depth in the exploration process. In order to use it to keep the diversity in the population as well, the *mutagenesis* operator is introduced. At the end of every generation, the mutagenesis operator chooses the N_w worst individuals which have survived to the next generation and changes the values of one of their genes in order to cover new zones of the search space (according to the information in the *gene matrix*). Specifically, for each of the N_w worst individuals in the population, one of the sub-ranges containing a zero value in the *GM* is selected randomly, and the value in the correspondent gene of the individual is updated according to a random value within the sub-rage boundaries. Afterwards, the zero in the *GM* is updated to a one and the process

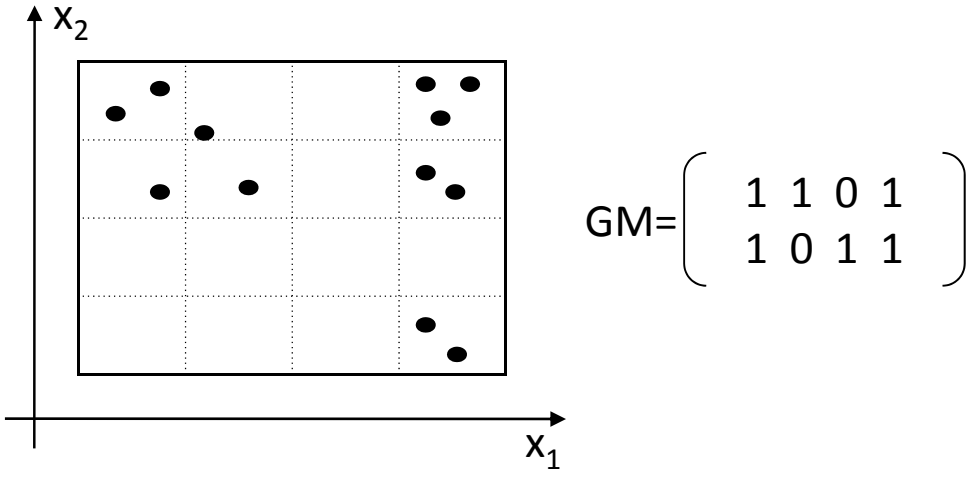


Figure 4.1: Gene Matrix example

continues until the rest of the N_w individuals have been modified or the GM is completely filled with ones. Figure 4.2 presents an example of this process.

The originally proposed termination criterion was based on the *gene matrix* values. Once all the sub-ranges of the variables in the search space had been covered (the *gene matrix* was completely filled with 1's) the algorithm continued for a certain number of generations (the dimensionality of the problem, n), in order to allow the evolutionary process to exploit the information from the last covered values, and afterwards it stopped. The heuristic to stop after a number of generations equal to the dimensionality of the problem obtained accurate results in a wide range of functions but proved not to be robust enough. An example can be seen using one of the test functions originally included in the algorithm's test-set: the first Schwefel function (eq. 4.1) in the search space of [-10, 10] using a dimensionality of 30 variables. Figure 4.3 shows a successful stopping situation for the function, while figure 4.4 shows an unsuccessful stop for the same function with the same algorithm configuration.

$$f(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \tag{4.1}$$

The termination criterion used is based only on the coverage of the variable space, using a simple heuristic to measure the effect on the objective function's space (the triggering of the stop after a number of generations equal to the problem dimensionality). The idea behind this criterion may be considered under the principles of distribution based stopping criteria (introduced in section 2.4.7). These criteria consider that, since all the different individuals in the population tend to converge to the optimum, distance measures regarding this fact may be used to determine whether the algorithm should be stopped (for instance measuring the standard deviation of the vectors in the population (Zaharie & Petcu, 2005)). ESLAT's stopping criterion uses this idea to build a stopping prevention mechanism, which attempts to prevent the situations which would trigger these criteria in the objective space. However, as figure 4.4 shows, this mechanism is not functional enough to properly detect stopping situations.

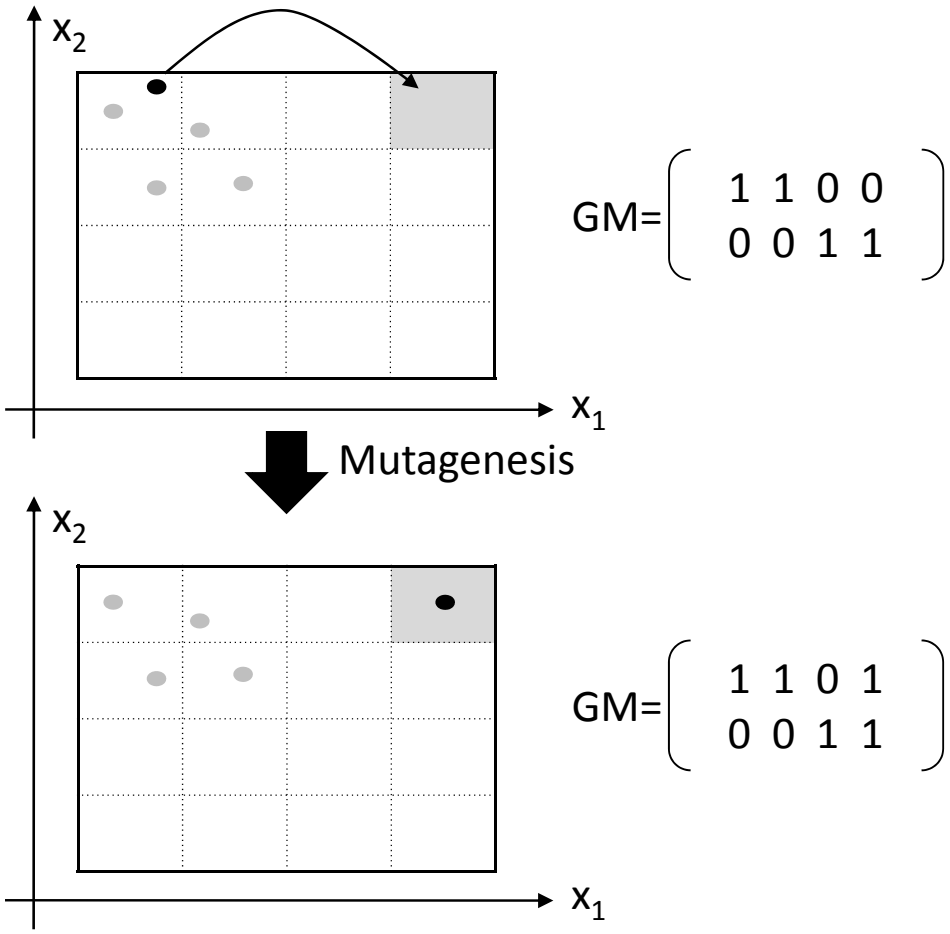


Figure 4.2: Mutagenesis operator example

Transformation operators and selection strategy

The recombination operator chosen for the technique is discrete, a $\rho - point$ crossover operator The value of ρ is chosen as the minimum between the dimensionality of the problem and 5, and it must be noted that only one child is produced in each recombination, according to the random partition points. This recombination operator is focused to exploration procedures, since exploitation will be mainly covered by the local search methods. By exchanging the different variable values of the parents (rather than arithmetically combining them) the algorithm seeks to cover more sub-ranges of the *gene matrix*.

The mutation operator is the standard for ES, evolving the mutation step (σ) along with the individuals, and choosing the following values for $\tau = 1/\sqrt{2\sqrt{n}}$ and $\tau' = 1/2\sqrt{n}$. The mutation step has also a minimum and maximum boundary, with values $\sigma_{min} = 1e^{-4}$ and $\sigma_{max} = 0.5 * d$ where d is the length of the search space. Seven mutated individuals are obtained for each individual to whom the mutation procedure is applied. In the main loop of the evolutionary technique, for each individual in the population, according to a recombination probability (ρ_r), the recombination or mutation procedure is applied (with a

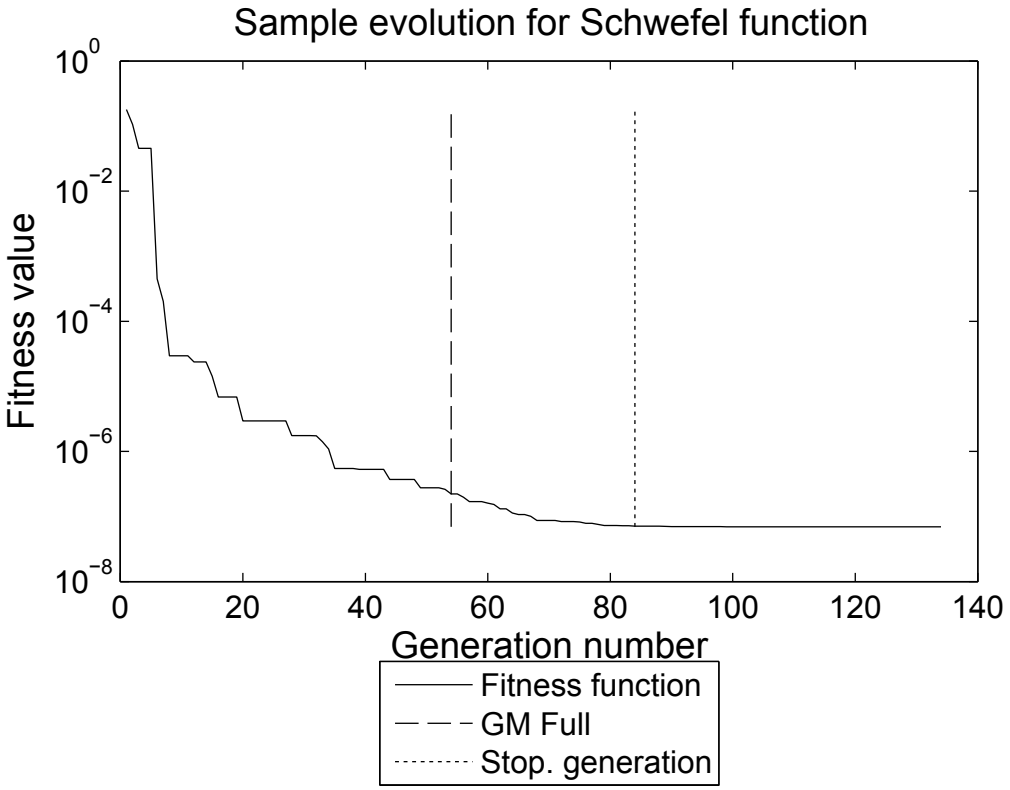


Figure 4.3: Schwefel function sucessful stop

noticeable impact in the number of offspring, since one produces one child and the other seven mutated children). The selection used is also standard: $(\mu/\rho + \lambda)$ selection. For more details on this standard operators, the reader may refer to sections 2.4.4 (crossover operators), 2.4.5 (mutation operators) and 2.4.6 (selection strategies).

Initialization procedure

The initialization process uses the scatter search diversification generation method (Laguna & Marti, 2003) to generate the initial population. This initialization divides each variable search space into four sub-ranges of equal size. Afterwards, the individuals of the initial population are created iteratively, choosing the variable sub-ranges with a probability inversely proportional to the number of solutions previously generated in that interval. Once all the sub-ranges for the different variables have been chosen, random values within the sub-ranges boundaries are chosen. It is interesting to highlight the similarities between this initialization method and the exploratory purpose of the *GM* and the mutagenesis operator focusing the initial population to cover a high number of sub-ranges in the *GM* and, therefore, speedup the evolutionary process.

The initialization section included in the state of the art chapter (section 2.4.2) also included some similar approaches, particularly that presented in (McKay et al., 1979), where the search space was divided into different sub-ranges and independent random initialization performed in each of them.

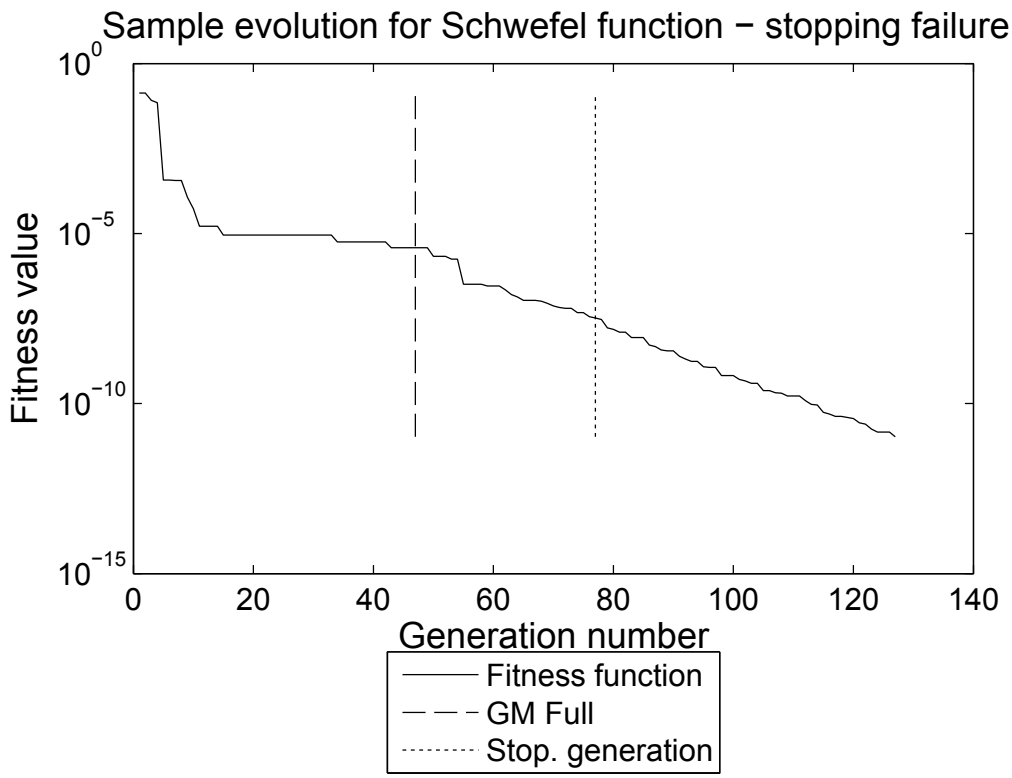


Figure 4.4: Schwefel function unsucessful stop

Intensification procedure

Finally, the so called *intensification procedure* is based on the application of two different derivative free local search techniques: the fminunc optimization procedure in Matlab and Kelley's improvement over the Nelder-Mead algorithm (Kelley, 2000). These techniques are applied as a step into the generational loop and also as a last step of the technique, once the algorithm reaches its stopping generation. Each of these techniques is applied *5n* generations, applying first the Nelder-Mead algorithm and the fminunc functions afterwards over its result. In the generational loop, this intensification is applied at most to two different individuals: the *best child* and the *most promising child*. The *best child* is the child who will update the current best individual in the following generation. The *most promising child* is the child with the greatest difference in fitness function value with his parent (either by mutation or recombination procedures). Figure 4.5 shows an overview of the different algorithm steps.

4.2.2 Introducing the R-ESLAT algorithm

R-ESLAT technique tries to deal with the analyzed drawbacks of the ESLAT algorithm. The first of these drawbacks is the lack of control over the search space (even some of the final solutions were found outside of it). An additional important feature is that the termination criterion focus on the exploration procedure, even though it does take into account exploitation in the form of the *n* generations which the evolutionary process continues before it is stopped, once the *GM* is full. Unfortunately, as shown in figure 4.4 and also in

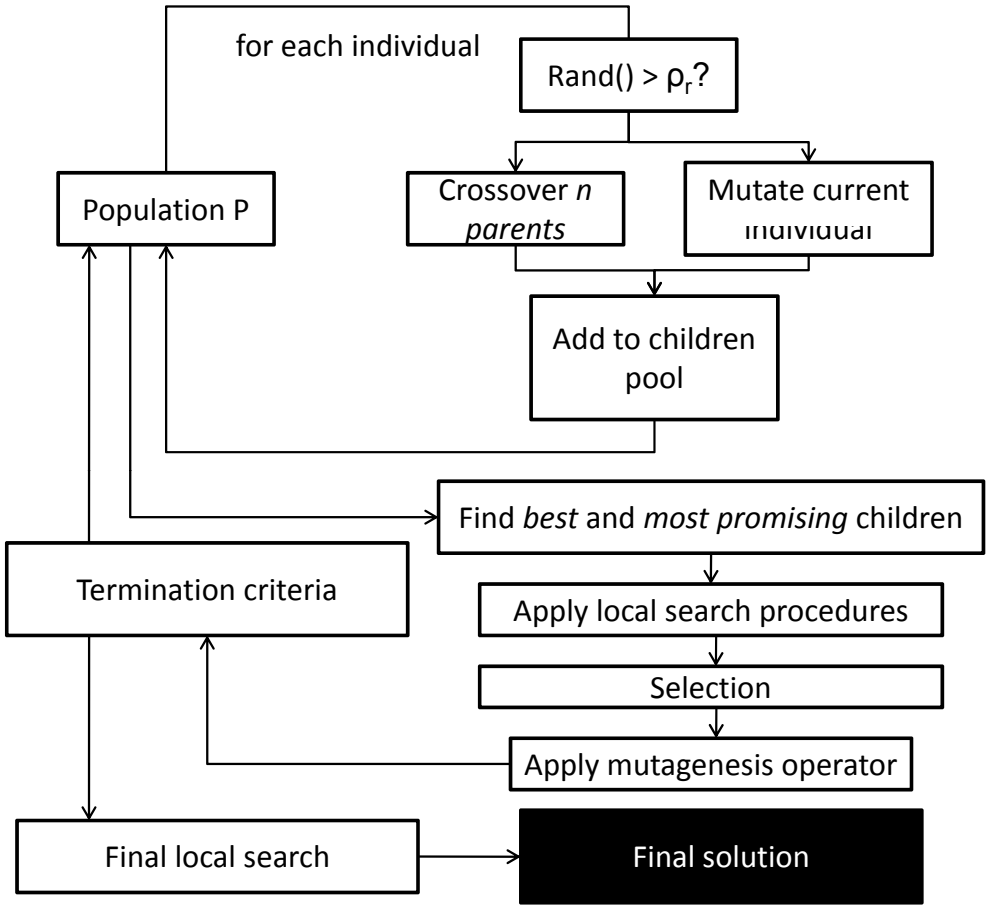


Figure 4.5: ESLAT algorithm overview

some figures in the original paper of the technique, this approach is not robust enough in many situations.

Another difficulty was the incomplete definition of the required configuration parameter values for the intensification process, which have a high impact in the quality of the obtained solution, since the exploitation process of the memetic algorithm relies on it. The objective of R-ESLAT is to completely define the previous processes, improve the overall robustness of the termination criterion and introduce the proper techniques to handle the boundaries of the search spaces. Regardless of its issues, the results of the original ESLAT technique already provided good quality results in comparison with CMAES in a certain number of functions (especially those with low dimensionality).

Search space control

The boundaries of the search space may be trespassed at two different steps of the algorithm: mutation and intensification processes. Two different approaches have been set for each of these processes: a repairing process is applied after the application of the mutation operator. If any of the mutated individuals have a variable value outside the search space, the mutation

results are rescaled considering the search space boundaries. This repairing procedure tries to disrupt the mutation procedure distribution as little as possible, while controlling the search space boundaries. On the intensification process, however, several boundary constrains may be unfulfilled at the same time, and the procedure to reach those values is not so clearly specified. For those reasons, a *death-penalty* approach is taken, leaving the intensified individual of the population in its original state. The joint application of these two processes guarantees that all the solutions will be contained within the problem variable boundaries.

Intensification process

The two local search algorithms proposed for ESLAT require different configuration parameters (and procedures to be set). The *fminunc* function of Matlab[1] provides default parameter values which may not be the most appropriate for the algorithm purpose. First of all, the concrete applied algorithm is defined by those parameters. Assuming the default algorithm choice parameters, the algorithm applied is the *Broyden-Fletcher-Goldfarb-Shanno* Quasi-Newton method (BFGS) with a cubic line search procedure (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). The algorithm requires certain tolerances to be set, both for the function and the variable values. Their default values are $1e^{-6}$, which are too high for the algorithm used. They have been reset to $1e^{-30}$. If the dimensionality of the problem was higher (the highest dimensionality included in the dataset is 30) the technique should be changed to the *interior-reflective* Newton method, involving the use of the preconditioned conjugate gradients method at each iteration (Coleman & Li, 1996). This technique would require a user-defined Hessian for the objective function as well.

The Nelder-Mead algorithm used in this chapter's proposal is based on Kelley's implementation included in (Kelley, 1999). An important feature of this technique is that, for a problem with n variables, it requires n+1 starting points to be applied. The ESLAT algorithm only defined one of such points (the *best* or *most promising* child) but did not include which technique should be used to choose the remaining n individuals. In fact, the choice of those individuals may lead to the exactly same populations for both intensifications, implying a waste of function evaluations. R-ESLAT chooses this population according to the first n individuals obtained in the children pool (which will be the children of the fittest individuals from the population) always excluding the element which is included in the complementary Nelder-Mead population (which means that, for the *best* child Nelder-Mead intensification initial set of individuals, the *most promising* child will never be included, and vice versa).

The original *best* child concept included only the best individual in the children pool when that children had a better fitness value than the previous best individual in the population. This meant that children with a good fitness value (but not good enough to become the new best individual in the population) who were exploring new regions of the search space did not get the chance to improve by means of the intensification process, and could disappear from the population, causing the algorithm to miss the chance to find a better minimum in that region. To prevent this behavior, the *best* children concept in R-ESLAT has been modified, and now it includes the child with the best fitness, regardless of whether it improves the previous best fitness in the population or not.

[1] Included in Matlab's optimization toolbox http //www mathworks com/products/optimization/

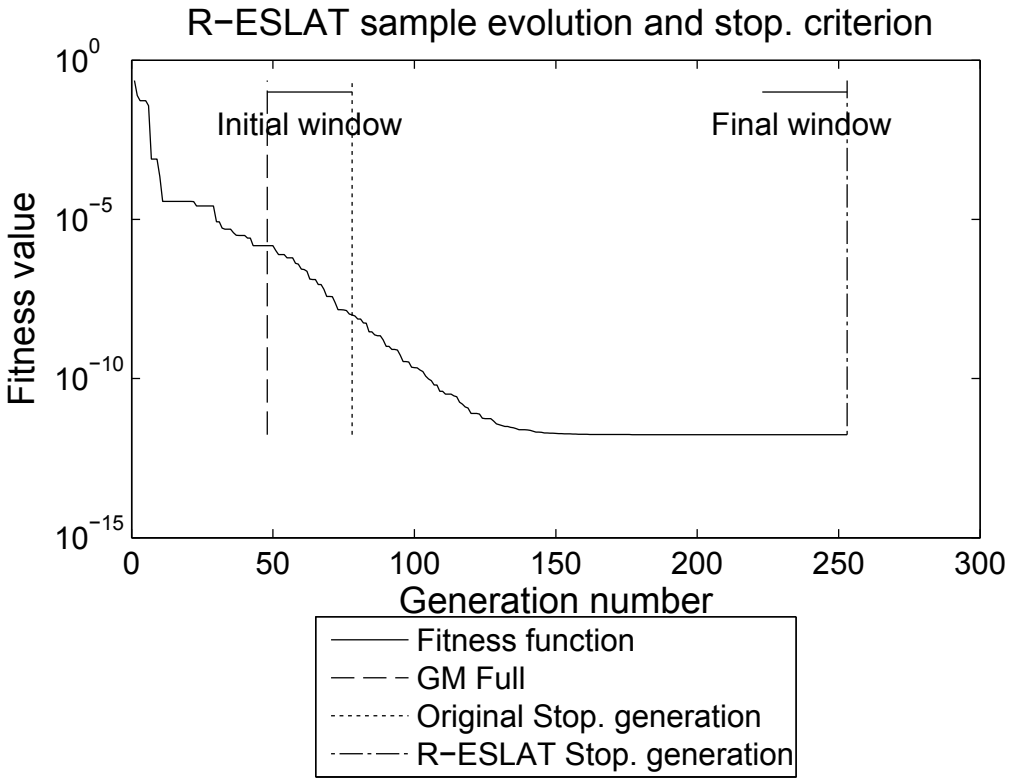


Figure 4.6: ESLAT algorithm stopping criterion example

Termination criterion

Finally, the termination criterion has to be modified in order to include exploitation considerations. The original termination criterion already included a certain *generational window* which established a number of generations before the final stopping took place. That generational window period was launched after the exploration had finished (the *GM* was full), and its purpose was to allow the exploitation process to use the information gathered in order to get the best possible final solution. To increase the stopping criterion robustness, we will analyze the value of the fitness function according to that window, and allow the continuation of the evolutionary process if the fitness function has improved its value in that given window. For every following generation, the generational window is moved one generation further, and the fitness values reanalyzed, until the fitness value remains constant in all the different generations contained in a certain generational window. Figure 4.6 shows an example of the improved termination criterion over the Schewefel function presented in eq. 4.1.

The improved stopping criterion is able to let the evolutionary process run until the exploitation process stagnates (unlike the original one, which stopped to algorithm while clear improvements over the fitness function were being made). Another important characteristic of this termination process is the size of the introduced *generational window*. In the ESLAT algorithm, that window is set to the dimensionality of the problem, n. However, in problems with low dimensionality, that size may be insufficient to gather enough information for an accurate stopping decision. An example can be seen in Beale's function, presented in equation 4.2.

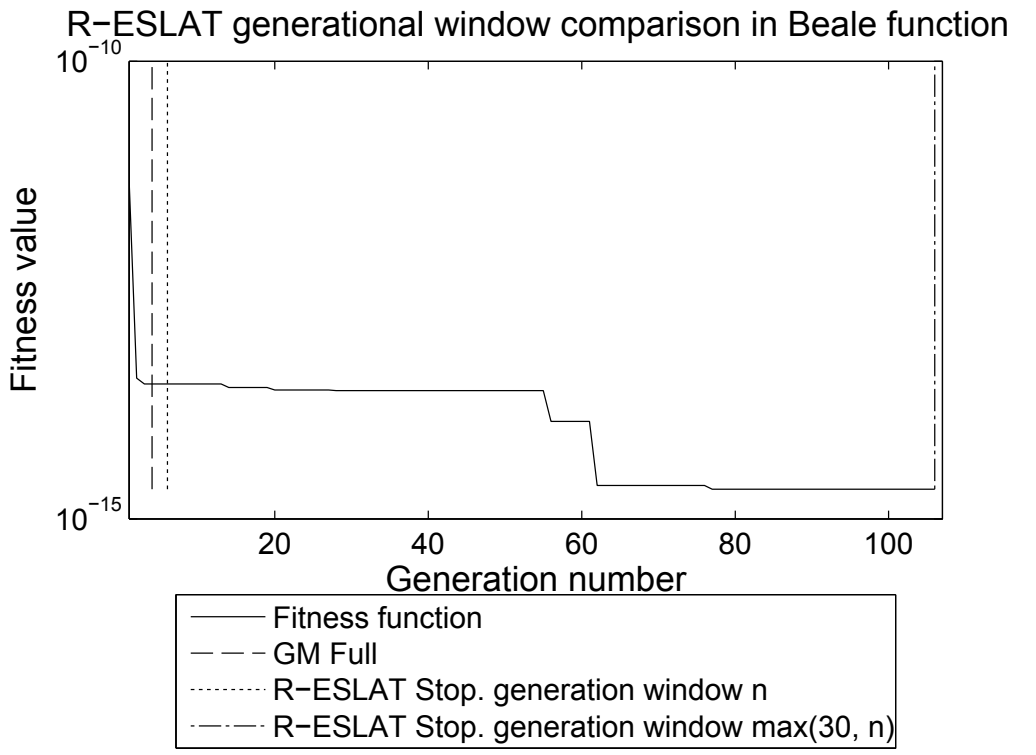


Figure 4.7: R-ESLAT generational window size comparison

$$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \tag{4.2}$$

Beale's function has a dimensionality of two, providing a very small generational window, which may lead to inaccurate stopping situations, as shown in figure 4.7. To improve the response of the stopping criterion, the generational window value in R-ESLAT is set to *max(30, n)* providing a fair amount of fitness evolution information for any problem, regardless of its dimensionality value. In figure 4.7 the effect of this increased generational window is shown, ending with a fitness value improved by almost several magnitude orders than the one obtained with the original generational window.

4.2.3 Experimental results

Through the previous sections, different parameters regarding ESLAT and R-ESLAT have been proposed and discussed, and will be used for the results presentation in this section. Table 4.1 shows an overview of them, where n is the problem dimensionality and d is the search space length. The obtained results with the proposed algorithm are compared to CMAES (Hansen et al., 2003), according to its Matlab implementation, version 3.54. The main parameters to be set in CMAES are the search space boundaries (set according to the concrete problem characteristics), the initial individual (chosen as a random value between the given boundaries) and the initial mutation step, set, according to the technique's author suggestions, as one third of the problem search space.

Table 4.1: R-ESLAT parameter values overview

Param.	Description	Value
μ	Population size	15
λ	Mutated children per individual	7
ρ	Mated parents	min(n, 5)
ρ_r	Recombination probability	0.5
σ_0	Initial mutation step parameter	3
σ_{min}	Min. mutation step parameter	$1e^{-4}$
σ_{max}	Max. mutation step parameter	0.5d
m	Sub-ranges in Gene Matrix	30
N_w	Worst ind. for mutagenesis	10
F_{tol}	Func. tolerance (intensification)	$1e^{30}$
Var_{tol}	Var. tolerance (intensification)	$1e^{30}$
gen_{win}	Generational window size	max(30, n)

The test-set used for the comparison contains 27 different functions with a wide set of different characteristics, regarding separability, dimensionality, search space, presence of local minima, etc. Table 4.2 details the functions used, along with their search space and the known minimum used. Chapter A includes the complete formulation of these functions, along with dimensionality, search space and their bidimensional representation, as an appendix to this chapter and particularly its experimental results.

The results over the dataset are presented in tables 4.3 (R-ESLAT), 4.4 (CMAES) and 4.5 (statistical significance). The fitness function presented is always the difference between the function value and the known global minimum. Thirty different independent executions were run for each problem. To determine the statistical significance of the results, the Wilcoxon test (Corder & Foreman, 2009) has been used. Table 6.1 shows that in 20 out of the 27 optimization functions in the dataset, the R-ESLAT algorithm has achieved better results than CMAES, in three cases the achieved differences were not statistically significant and in four cases CMAES obtained better results than R-ESLAT.

One interesting feature to compare is the difference in the standard deviation values. In CMAES the standard deviation is relatively high when compared to the mean value, meaning that the results of the technique can be very different to each other when run a single time. This feature is present even in some of those cases when the statistical test did not find enough significance in the results difference. Figure 4.8 shows an example of this behavior on Branin's function (f5). In that figure, it is shown that in 6 out of the 30 independent runs, CMAES provides significantly worse results that R-ESLAT, being similar in the remaining ones.

Throughout this work, the two different procedures involved in an optimization process (exploration and exploitation) have been highlighted and analyzed independently regarding the treatment which they receive in the R-ESLAT algorithm. In the results presentation in tables 4.3-4.4, the performance of the exploration process can be seen particularly in functions 3, 10 and 26, where R-ESLAT technique is able to find the global minimum in the 30 different runs (while CMAES cannot reach those global minima in any of the three different test functions).

Table 4.2: Test-set functions overview

id	Name	n	min. bound.	max. bound.
f1	Ackley	30	-15	30
f2	Beale	2	-4,5	4,5
f3	Bohachecsky	2	-100	100
f4	Booth	2	-10	10
f5	Branin	2	-5	15
f6	Colville	4	-10	10
f7	Dixon-Price	30	-10	10
f8	Easom	2	-100	100
f9	Goldstein-Price	2	-2	2
f10	Griewank	30	-600	600
f11	Hartmann	6	0	1
f12	Hump	2	-5	5
f13	Levy	30	-10	10
f14	Matyas	2	-10	10
f15	Michalewicz	10	0	pi
f16	Perm	30	-30	30
f17	Powell	28	-4	5
f18	Power Sum	4	0	4
f19	Rastrigin	30	-5,12	5,12
f20	Rosenbrock	30	-5	10
f21	Schwefel	30	-500	500
f22	Shekel	4	0	10
f23	Shubert	2	-10	10
f24	Sphere	30	-5,12	5,12
f25	Sum Squares	30	-10	10
f26	Trid	10	-100	100
f27	Zakharov	30	-5	10

Table 4.3: R-ESLAT results

id	Fitness		F. evals	
	Mean	Std	Mean	Std
f1	4,44E-15	4,01E-30	2,88E+05	1,05E+04
f2	7,36E-15	9,78E-15	1,29E+04	4,79E+03
f3	0,00E+00	0,00E+00	8,31E+03	9,91E+01
f4	2,89E-16	4,27E-16	9,64E+03	2,21E+03
f5	3,58E-07	8,95E-16	8,47E+03	1,89E+03
f6	8,85E-14	1,16E-13	3,75E+04	4,26E+04
f7	6,67E-01	8,40E-16	2,52E+05	2,76E+04
f8	4,00E-01	4,98E-01	7,87E+03	3,96E+03
f9	6,81E-16	6,97E-16	1,02E+04	2,37E+03
f10	0,00E+00	0,00E+00	3,01E+05	7,84E+03
f11	1,99E-06	1,39E-15	2,22E+04	3,79E+03
f12	4,65E-08	1,54E-16	8,68E+03	1,97E+03
f13	4,83E-20	2,64E-19	2,01E+05	1,67E+04
f14	3,85E-21	7,96E-21	9,83E+03	2,48E+03
f15	8,41E-02	7,67E-02	7,23E+04	1,26E+04
f16	8,62E+81	8,84E+81	1,76E+05	1,79E+05
f17	2,37E-12	1,97E-12	4,47E+05	1,07E+05
f18	1,13E-06	3,75E-06	2,93E+04	1,01E+04
f19	3,65E+00	3,02E+00	1,77E+05	5,08E+04
f20	3,98E-11	1,73E-11	4,81E+05	4,42E+05
f21	5,48E+02	1,82E+02	1,70E+05	2,08E+04
f22	5,92E-16	1,26E-15	2,06E+04	4,20E+03
f23	5,08E-12	1,51E-11	1,13E+04	2,98E+03
f24	5,22E-23	4,75E-23	1,42E+05	1,20E+04
f25	1,10E-20	2,17E-20	2,65E+05	2,40E+04
f26	0,00E+00	0,00E+00	4,97E+04	4,97E+03
f27	1,20E-14	9,27E-15	6,23E+05	7,09E+05

Table 4.4: CMAES results

id	Fitness		F. evals	
	Mean	Std	Mean	Std
f1	8,34E+00	4,07E+00	7,40E+03	1,12E+03
f2	5,08E-02	1,93E-01	6,16E+02	9,41E+01
f3	6,08E-02	1,58E-01	6,59E+02	4,09E+01
f4	6,75E-16	2,14E-15	5,71E+02	4,41E+01
f5	4,62E-01	9,39E-01	6,11E+02	5,30E+01
f6	4,43E-16	5,43E-16	2,32E+03	7,18E+02
f7	6,67E-01	1,20E-15	1,07E+04	1,09E+03
f8	1,00E+00	0,00E+00	8,20E+00	1,10E+00
f9	8,10E+00	2,47E+01	7,44E+02	4,90E+02
f10	1,80E-03	5,59E-03	9,43E+03	2,10E+02
f11	3,97E-02	5,72E-02	1,79E+03	2,58E+02
f12	4,65E-08	3,67E-16	5,64E+02	3,81E+01
f13	7,86E-01	1,12E+00	8,29E+03	5,72E+02
f14	1,53E-16	3,50E-16	5,23E+02	2,84E+01
f15	2,27E+00	7,35E-01	5,82E+03	2,88E+03
f16	1,50E+85	6,86E+85	2,28E+05	1,46E+05
f17	1,05E-11	1,18E-11	4,40E+04	3,01E+03
f18	8,72E-12	1,07E-11	1,88E+04	5,89E+03
f19	6,54E+01	2,16E+01	1,31E+04	4,25E+03
f20	5,32E-01	1,38E+00	4,91E+04	2,46E+03
f21	5,34E+03	5,79E+02	2,71E+04	2,50E+03
f22	5,11E+00	3,75E+00	1,42E+03	4,45E+02
f23	5,57E+01	7,51E+01	1,05E+03	1,96E+02
f24	1,17E-15	4,46E-16	6,74E+03	1,53E+02
f25	2,03E-15	9,09E-16	9,12E+03	2,53E+02
f26	6,06E-14	2,31E-13	3,33E+03	1,96E+02
f27	3,80E-15	1,75E-15	1,75E+04	3,59E+02

Table 4.5: Wilcoxon test results

id	p-value	Significantly best technique
f1	1,21178E-12	R-ESLAT
f2	3,82972E-05	R-ESLAT
f3	0,002785834	R-ESLAT
f4	0,043583548	R-ESLAT
f5	0,722098485	-
f6	8,10136E-10	CMAES
f7	8,64808E-05	R-ESLAT
f8	8,4555E-07	R-ESLAT
f9	1,47443E-07	R-ESLAT
f10	1,19996E-12	R-ESLAT
f11	0,142835794	-
f12	0,282958147	-
f13	2,97474E-11	R-ESLAT
f14	3,01986E-11	R-ESLAT
f15	3,00287E-11	R-ESLAT
f16	1,06657E-07	R-ESLAT
f17	0,002052334	R-ESLAT
f18	3,01986E-11	CMAES
f19	2,89542E-11	R-ESLAT
f20	1,10393E-06	R-ESLAT
f21	2,9991E-11	R-ESLAT
f22	1,22758E-10	R-ESLAT
f23	0,639982441	-
f24	3,01986E-11	R-ESLAT
f25	3,01986E-11	R-ESLAT
f26	0,160741998	-
f27	1,99628E-05	CMAES

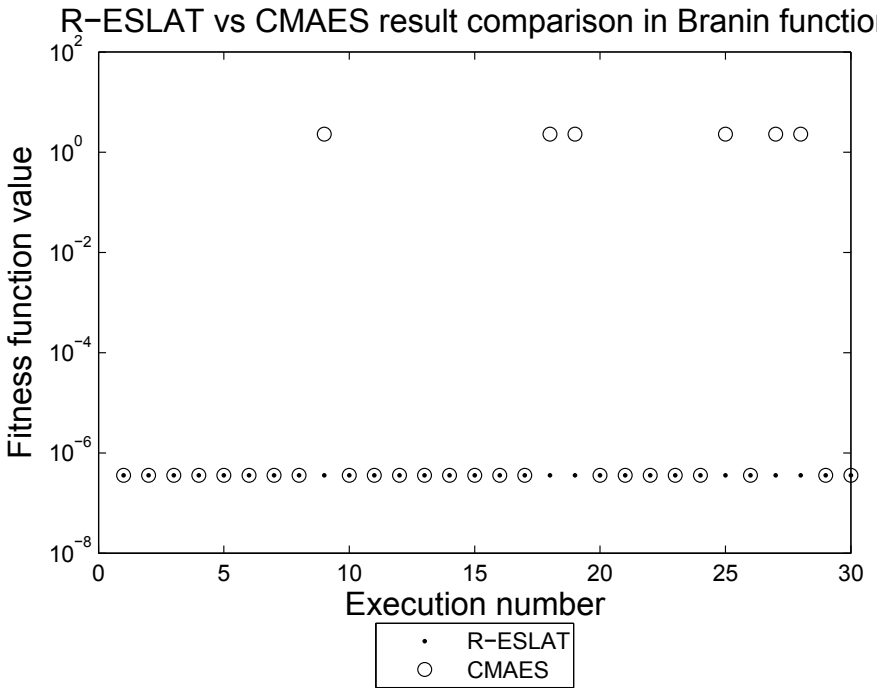


Figure 4.8: R-ESLAT and CMAES results comparison over the 30 independent runs performed on the Branin's function test problem

The exploration process can be hard to evaluate independently. For this purpose, Easom function may be the most appropriate function in the dataset due to its particular topology (which provides almost no information which can be exploited for most of its search space). Its formulation is presented in eq. 4.3, and its representation in figure 4.9.

$$f_8(x) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2} \tag{4.3}$$

The results of the two compared techniques are presented in figure 4.10. As shown in it, R-ESLAT is able to find the minimum (or at least its location zone, if not the global minimum) in 18 out of the 30 runs, while CMAES is not able to do so in any of the 30 runs. This shows the capacities of the exploration process in R-ESLAT. It can be highlighted from the results in table 4.4, function f8, that the stopping criterion of CMAES, lacking guidance information, is triggered after a very small number of function evaluations.

Obviously, the cost of the results for the presented algorithm is shown in the number of function evaluations performed, several orders over CMAES (caused by the application of local search techniques requiring a high number of function evaluations in R-ESLAT and the derandomized search performed by CMAES which allows the technique to perform a low number of function evaluations).

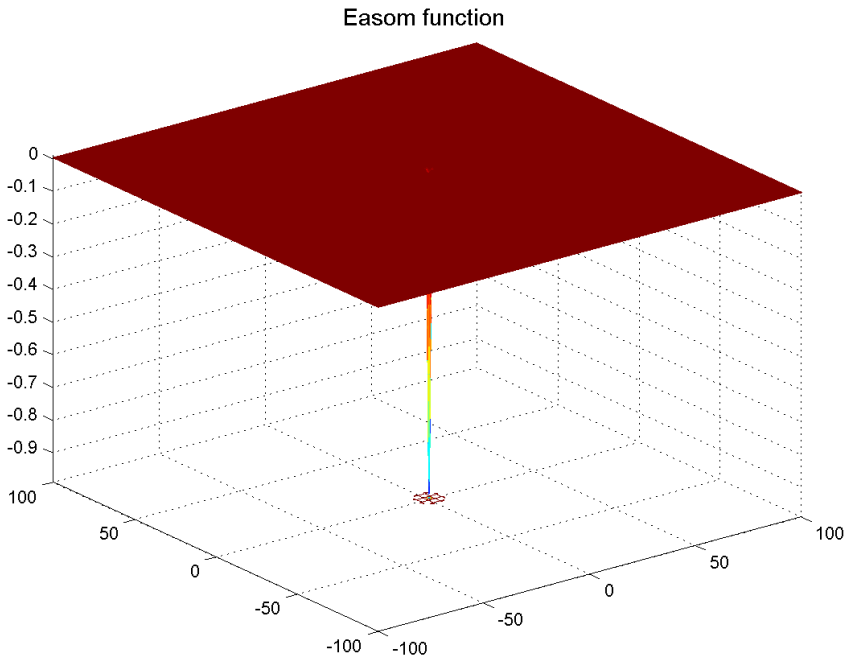


Figure 4.9: Easom function

4.3 Mutagenesis as a diversity enhancer and preserver in evolutionary strategies

Section 4.2.2 has presented the R-ESLAT technique, which introduced a modified stopping criterion based on the mutagenesis operator. The results presented have showed that the stopping criterion is effective as part of the integrated memetic approach, based on an enhanced exploitation procedure. The objective of this section is to extract this procedure from the memetic schema, remark its diversity effect as an early stopping prevention mechanism, performing the required changes, and establish its isolated effect comparing the technique versus a canonical evolutionary strategy.

4.3.1 Mutagenesis as an independent transformation operator

Mutagenesis and the gene matrix artifact have been detailed, according to their use in R-ESLAT, in section 4.2.1. As seen in figure 4.1, the gene matrix basically covers which sub-ranges have been covered for each of the variables, and mutagenesis (figure 4.2) forces changes for specific gene values to sub-ranges not previously covered.

The initial proposal was based on a gene matrix with a fixed size, which is iteratively filled with 1's, until this matrix is completely filled with 1's, which triggers the secondary stopping approach, based on the value of the best individual in the different populations. This approach has a series of issues: establishing the a-priori size of the gene matrix and the lack of diversity enhancement during the final phase of the evolutionary process (once the secondary stopping criterion has been triggered). This process is summarized in figure 4.11

The novel gene matrix proposal is focused on diversity enhancement, rather than its

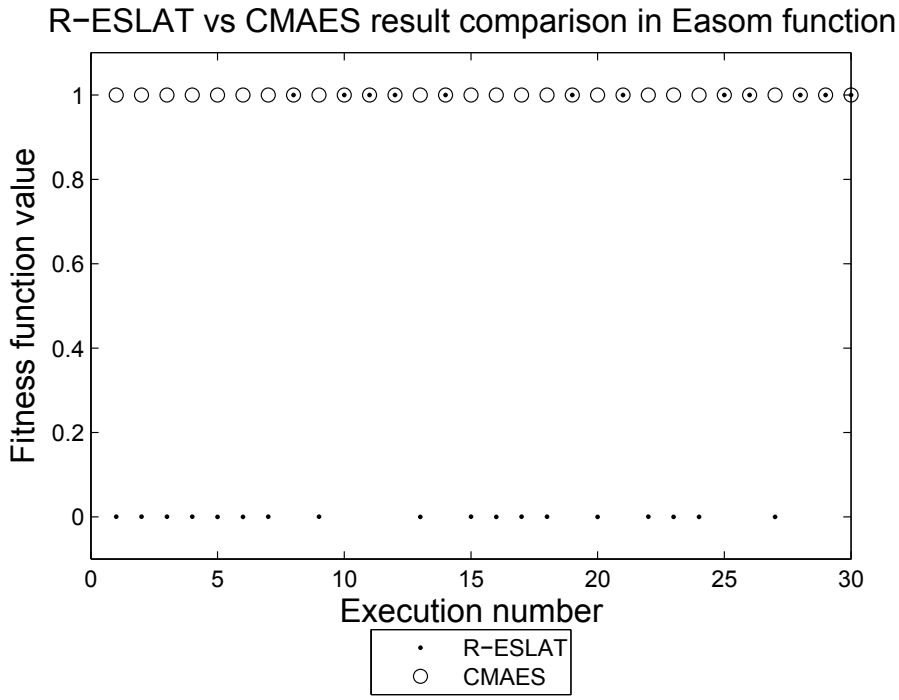


Figure 4.10: R-ESLAT and CMAES results comparison over the 30 independent runs performed on the Easom's function test problem

application as a termination criterion (which is still performed). To do so, an initial number of subranges is set a-priori. Once the gene matrix is filled with ones at a certain generation, it is *restarted*, reinitializing it with zeros and updating it with the individuals in the population which caused this reinitialization. Every time the gene matrix is reinitialized, its number of contained subranges is doubled. This mechanism achieves a constant diversity enhancement and also a more thorough coverage of the search space as the algorithm progresses, depending on the dimensionality of the problem faced.

The mutagenesis procedure has also been reviewed. As previously explained, it originally introduced a certain number of modifications on the worst individuals of the population, changing concrete values from the chromosome to unexplored subranges of the chosen gene. This behavior may not introduce enough diversity in a population heavily dominated by the best individual, so an additional probability is added to the algorithm configuration: p_{rm}, random mutagenesis probability. According to this probability, mutagenesis may generate a random individual covering the chosen subrange instead of modifying just one gene from one of the worst individuals in the population.

Additional controls have also been added to mutagenesis. If an individual has covered a new subrange in current generation, it is never changed any further by the mutagenesis procedure, regardless of its rank. This allows the new information introduced during the evolutionary cycle to survive at least one generation, in order to give the new individual the chance to procreate and mutate before any directed change is applied to it. This also implies a change in the mutagenesis configuration. Instead of N_w changed individuals, the user configures a more versatile N_c parameter, establishing the number of new subranges

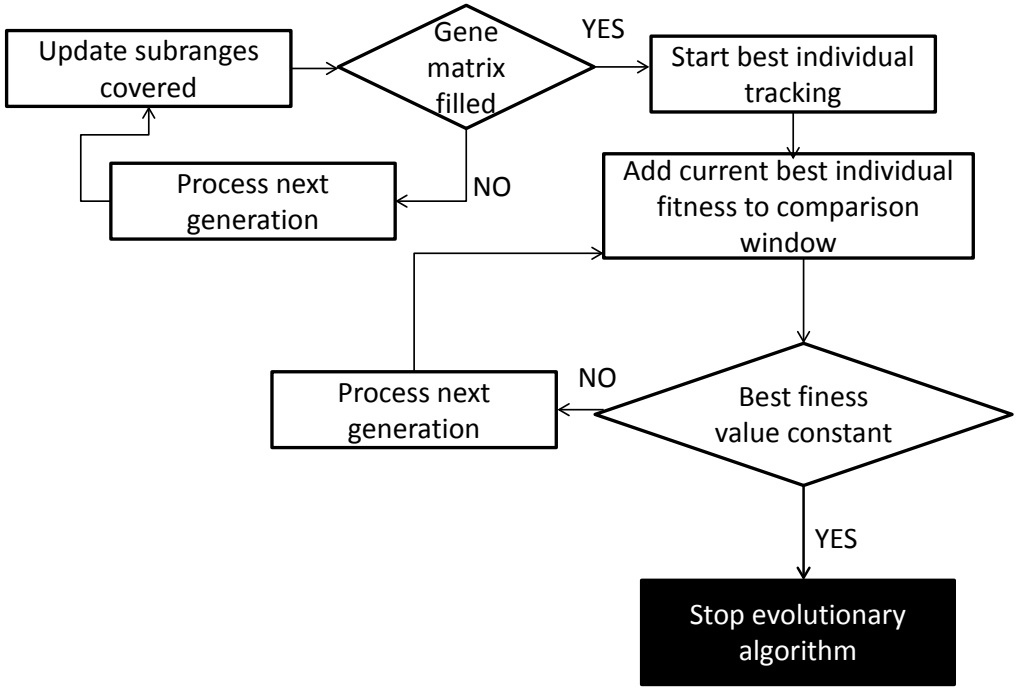


Figure 4.11: R-ESLAT stopping management overview

covered each generation. If the evolutionary cycle covers the required number of changes, no mutagenesis is applied. In other case, the worst individuals (as many as required in order to cope with the desired N_c changes) are picked to go through the mutagenesis procedure.

Figure 4.12 includes the flow diagram describing the presented mutagenesis procedure, along with the growing gene matrix management for the stopping and diversity enhancement procedure

Finally, the stopping criterion used in R-ESLAT implied that the best fitness was repeated over a certain window of generations. This exact repetition may be too strict for a stopping criterion, since very small changes in fitness values (which might even be affected by the representation precision) would lead to a continuation in the evolutionary algorithm once the search process had stagnated regarding all practical purposes. For these reasons this exact comparison was changed to the comparison quotient presented in equation 4.4, which provides a more flexible mechanism to control the relevance of the changes.

$$\frac{previous_{best} - current_{best}}{previous_{best}} \leq Improvement_{factor} \tag{4.4}$$

The proposed algorithm is based on the following principles: a stopping prevention mechanism based on the exploration enhancement provided by a gene matrix which adapts its size according to the search deph, a mutagenesis operator which is applied only as a complement to the search when required (according to the number of subranges covered by the transformation operators on each generation) and allows its introduced information to remain in the evolutionary cycle at least for a generation and, finally, a stopping criterion which is based on the tracking and stagnation detection of the best individual fitness value, considering the exploration enhancements introduced by the previous approaches.

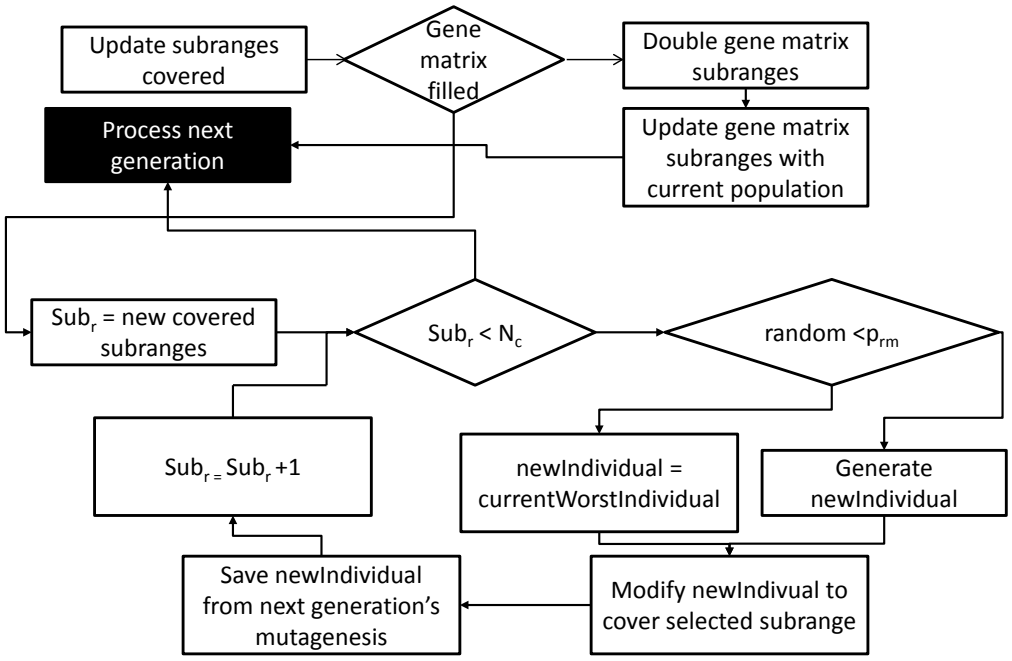


Figure 4.12: Diversity enhancement procedure overview

4.3.2 Experimental validation

The same set of 27 single optimization functions used for the validation of R-ESLAT is used, in this case, for the validation of the presented technique versus a canonical evolutionary strategy. The overview of the dataset is included in table 4.2, while appendix A includes a thorough description.

Several parameters (according to their description included in the previous section) have to be established for the proposed technique, which are presented in table 4.6. As included in that table, four different population sizes are used to cover the comparison of the two different techniques. The complete results for the different population sizes are presented in tables 4.8-4.11. A comparative presentation of those results is presented in table 4.3.2 and figure 4.13. Following (García et al., 2009), the individual comparison for the different test functions is performed according to parametric and non-parametric tests. The normality test used is the Shapiro-Wilk test, the parametric test is Student's t-test and the non-parametric test is Wilcoxon signed-rank test. The statistical best results are provided according to the t-test if the data follows a normal distribution and according to the non-parametric test otherwise. Fifty iterations have been run in order to establish the statistical significance of the results.

To test the final performance comparison, a Wilcoxon rank-sum test is carried out over the mean results for the twenty-seven functions and the four considered population sizes. The p-value obtained is 0.0275, which implies that with a significance level as low as 3% (lower than the usual 5% considered for these tests) the proposed gene matrix diversity enhancer allows evolution strategies to perform better.

Analyzing the individual results, the effectiveness of the diversity enhancement is, in general, more representative at lower population sizes (where the risk of falling into local

Table 4.6: Experimental configuration

Parameter	Description	Value
μ	Population size	5, 10, 15, 30
$init_{sr}$	Initial subranges	10
min_{sr}	Minimum subranges covered per generation	$\mu/5$
p_{rm}	random mutagenesis probability	0.5
l_f	Improvement factor	1E-05

Table 4.7: Results comparison for the different considered population sizes

Population size	Statistical Best	Statistical Worst	Best
5	7	3	19
10	7	8	14
15	10	6	15
30	3	10	13

optima is higher and the exploration capabilities are reduced) but, at the same time, since the number of required changes per generation are configured as a certain percentage of the population, the use of the gene matrix is more accused on higher population sizes. The balance between these two factors determines the effectiveness of the mutagenesis changes. This is reflected in the variable number of significant best and worst results obtained for the different population sizes.

Finally, regarding the individual analysis of the results for the different test functions, it must be noted that the non-parametric tests do not seem to be able to properly measure some behavior differences (due to their zero median null hypotheses). This can be seen, for instance, in table 4.8, function f14, where, even though the mean value obtained by the evolution strategy using mutagenesis is several orders of magnitude better, the Wilcoxon test does not determine it to be the best. It must be noted, though, that Wilcoxon test assumes zero skewness, even tough its is generally applied in the literature, and thus in this chapter, without the proper cheks for this fact (which tends not to be true in evolutionary algorithm results, due to the effect of local optima) . This points to the requirement of mean based statistical tests not requiring normality distribution over their measures to perform quality comparisons between algorithms.

A closer inspection is required for the detailed incomparable example. This is provided in figures 4.14 and 4.15. In figure 4.14, the outlier provided by the canonical approach makes it hard to assess the performance of the algorithms. Figure 4.15 provides the same results removing the outlier. As can be seen, the technique using the gene matrix is much more robust to early stagnation of the algorithm, although the computational effort spent in the exploration process hampers its exploitation capabilities, making it reach higher final fitness values. This highlights the tradeoff established by the proposed technique: more robust results in terms of success rate which may require a higher number of function evaluations to reach their best final result.

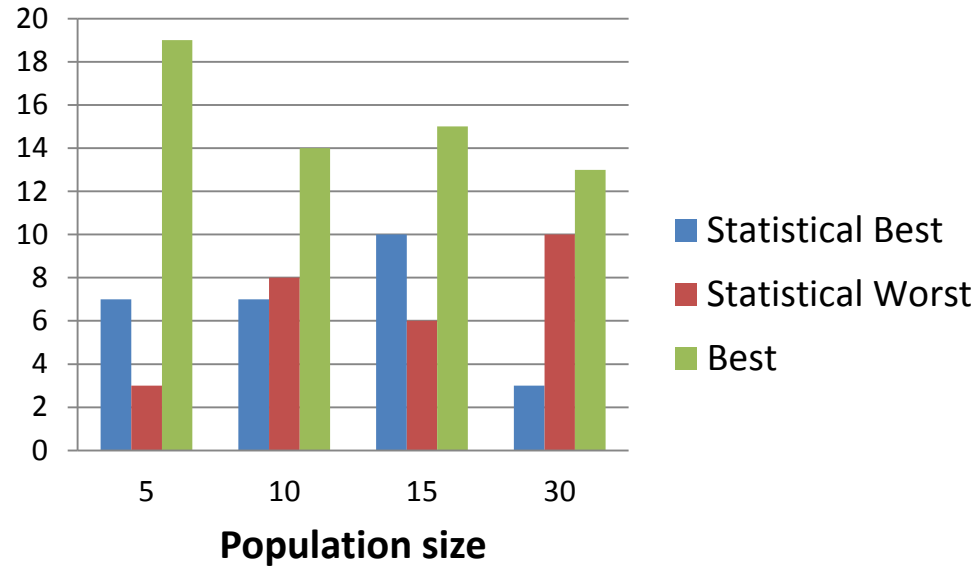


Figure 4.13: Results comparison at different population sizes

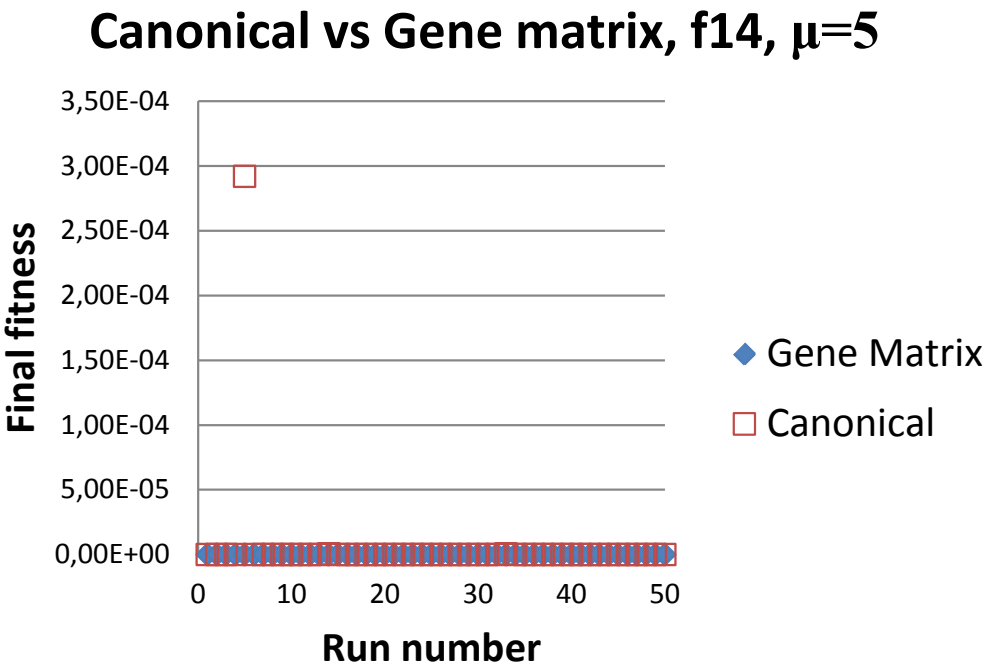


Figure 4.14: Results comparison with function f14 and population size five

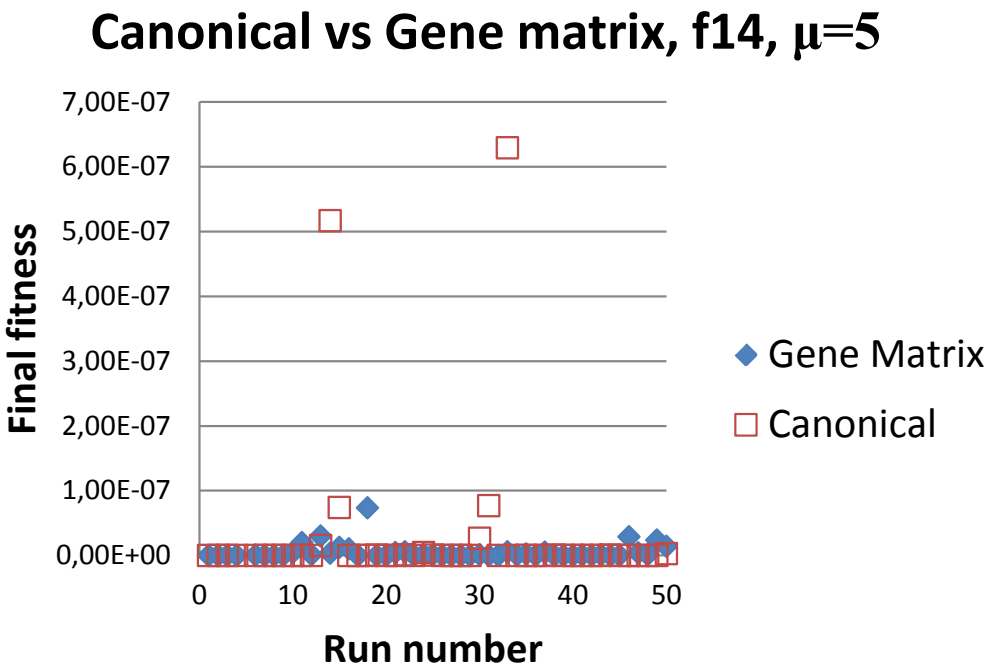


Figure 4.15: Results comparison with function f14 and population size five removing an outlier

Table 4.8: Test-set functions overview and results for population size five

id	Gene Matrix results		Canonical results		Statistical tests results			Techniques comparison	
	std	mean	std	mean	normal	ttest	wilcoxon	statistical best	best
f1	1,41E+00	1,45E+00	1,54E+00	1,95E+00	1	0	0		Gene Matrix
f2	3,05E 02	1,51E 01	3,04E 01	4,13E 01	1	1	1	Gene Matrix	Gene Matrix
f3	6,12E 02	1,78E 01	8,31E 03	5,85E 02	1	1	1	Canonical	Canonical
f4	8,19E 08	2,62E 07	1,53E 07	5,86E 07	1	0	0		Gene Matrix
f5	4,58E 07	4,44E 07	4,79E 02	3,26E 01	1	0	0		Gene Matrix
f6	6,04E 01	1,39E+00	1,02E+00	2,87E+00	1	0	0		Gene Matrix
f7	2,45E+00	1,98E+00	1,80E+00	1,46E+00	1	1	0		Canonical
f8	8,83E 01	3,21E 01	9,50E 01	1,78E 01	1	0	0		Gene Matrix
f9	2,16E+00	7,40E+00	9,92E+00	2,66E+01	1	0	0		Gene Matrix
f10	1,21E 01	2,15E 01	1,29E 01	1,97E 01	1	0	0		Gene Matrix
f11	3,82E 02	5,62E 02	3,36E 02	5,39E 02	1	0	0		Canonical
f12	1,63E 02	1,15E 01	6,03E 06	2,96E 05	1	0	0		Canonical
f13	1,14E 01	3,50E 01	4,76E 01	1,31E+00	1	1	1	Gene Matrix	Gene Matrix
f14	5,17E 09	1,23E 08	5,86E 06	4,13E 05	1	0	0		Gene Matrix
f15	5,81E 01	3,35E 01	1,01E+00	5,01E 01	0	1	1	Gene Matrix	Gene Matrix
f16	2,59E+85	4,57E+85	1,35E+86	1,56E+86	1	1	1	Gene Matrix	Gene Matrix
f17	1,08E 02	1,97E 02	6,51E 03	1,11E 02	1	1	1	Canonical	Canonical
f18	4,05E 02	3,24E 02	1,24E 01	4,16E 01	1	0	0		Gene Matrix
f19	2,35E+01	1,20E+01	4,25E+01	1,86E+01	0	1	1	Gene Matrix	Gene Matrix
f20	9,90E+01	1,66E+02	7,72E+01	1,21E+02	1	0	0		Canonical
f21	7,69E+02	4,72E+02	2,35E+03	4,07E+02	0	1	1	Gene Matrix	Gene Matrix
f22	5,54E+00	3,27E+00	5,28E+00	3,41E+00	0	0	0		Canonical
f23	2,72E 02	1,84E 01	2,65E+00	1,87E+01	1	0	0		Gene Matrix
f24	5,14E 05	2,01E 04	1,46E 05	5,58E 05	1	0	1	Canonical	Canonical
f25	2,84E 04	1,03E 03	1,53E 03	9,70E 03	1	0	0		Gene Matrix
f26	6,00E 02	1,36E 01	1,07E 01	4,56E 01	1	0	1	Gene Matrix	Gene Matrix
f27	1,97E+01	4,37E+01	2,25E+01	4,39E+01	1	0	0		Gene Matrix

Table 4.9: Test-set functions overview and results for population size ten

id	Gene Matrix results		Canonical results		Statistical tests results			Techniques comparison	
	std	mean	std	mean	normal	ttest	wilcoxon	statistical best	best
f1	5,28E 01	9,46E 01	3,95E 01	8,70E 01	1	1	1	Canonical	Canonical
f2	1,52E 02	1,08E 01	2,18E 01	3,82E 01	1	1	1	Gene Matrix	Gene Matrix
f3	8,26E 03	5,84E 02	4,96E 08	2,35E 07	1	0	1	Canonical	Canonical
f4	2,20E 10	4,98E 10	8,36E 08	5,33E 07	1	0	0		Gene Matrix
f5	4,62E 02	3,26E 01	9,23E 02	4,57E 01	1	0	0		Gene Matrix
f6	1,00E 02	2,04E 02	5,69E 03	1,08E 02	1	0	1	Canonical	Canonical
f7	1,64E+00	1,44E+00	1,41E+00	1,44E+00	0	0	0		Canonical
f8	8,00E 01	4,04E 01	8,01E 01	4,02E 01	1	0	0		Gene Matrix
f9	2,93E 10	9,06E 10	1,62E+00	1,15E+01	1	0	0		Gene Matrix
f10	4,54E 02	4,63E 02	4,45E 02	5,03E 02	1	0	0		Canonical
f11	2,38E 02	4,82E 02	5,72E 02	6,02E 02	1	1	1	Gene Matrix	Gene Matrix
f12	4,74E 08	3,61E 09	4,66E 08	3,80E 10	1	0	1	Canonical	Canonical
f13	5,77E 02	2,13E 01	8,02E 02	4,00E 01	1	0	1	Gene Matrix	Gene Matrix
f14	3,10E 10	1,80E 09	5,41E 09	2,33E 08	1	0	0		Gene Matrix
f15	4,38E 01	2,23E 01	6,60E 01	3,72E 01	0	1	1	Gene Matrix	Gene Matrix
f16	6,85E+84	1,61E+85	5,18E+85	8,64E+85	1	1	1	Gene Matrix	Gene Matrix
f17	3,09E 04	2,17E 04	1,92E 04	1,59E 04	1	1	1	Canonical	Canonical
f18	1,42E 02	2,49E 02	2,20E 02	3,70E 02	1	0	0		Gene Matrix
f19	1,78E+01	6,85E+00	2,66E+01	8,63E+00	0	1	1	Gene Matrix	Gene Matrix
f20	4,65E+01	6,58E+01	3,19E+01	5,09E+01	1	0	0		Canonical
f21	6,47E+02	3,83E+02	1,91E+03	3,69E+02	0	1	1	Gene Matrix	Gene Matrix
f22	4,69E+00	3,76E+00	4,67E+00	3,74E+00	1	0	0		Canonical
f23	2,08E 05	1,47E 04	2,12E 07	7,99E 07	1	0	0		Canonical
f24	4,38E 08	1,89E 07	8,87E 09	2,97E 08	1	0	1	Canonical	Canonical
f25	9,31E 07	2,54E 06	4,63E 07	1,33E 06	1	0	1	Canonical	Canonical
f26	8,18E 05	1,06E 04	2,13E 05	4,09E 05	1	1	1	Canonical	Canonical
f27	6,65E 02	2,46E 01	2,34E 02	8,27E 02	1	0	1	Canonical	Canonical

Table 4.10: Test-set functions overview and results for population size fifteen

id	Gene Matrix results		Canonical results		Statistical tests results			Techniques comparison	
	std	mean	std	mean	normal	ttest	wilcoxon	statistical best	best
f1	9,79E 02	3,43E 01	2,18E 02	1,35E 01	1	0	1	Canonical	Canonical
f2	1,52E 02	1,08E 01	4,60E 02	1,83E 01	1	0	0		Gene Matrix
f3	3,90E 09	2,22E 08	6,50E 09	4,36E 08	1	0	1	Gene Matrix	Gene Matrix
f4	3,56E 10	2,29E 09	2,05E 09	1,34E 08	1	0	0		Gene Matrix
f5	3,58E 07	2,63E 10	9,23E 02	4,57E 01	1	0	1	Gene Matrix	Gene Matrix
f6	1,48E 01	1,05E+00	3,81E 02	2,54E 01	1	0	1	Canonical	Canonical
f7	1,78E+00	1,52E+00	1,59E+00	1,49E+00	0	0	0		Canonical
f8	5,20E 01	5,05E 01	3,71E 01	4,77E 01	1	1	0		Canonical
f9	5,40E 01	3,82E+00	1,62E+00	1,15E+01	1	0	1	Gene Matrix	Gene Matrix
f10	3,08E 02	3,16E 02	4,04E 02	3,72E 02	0	0	0		Gene Matrix
f11	3,10E 02	5,28E 02	4,77E 02	5,90E 02	1	0	0		Gene Matrix
f12	4,65E 08	4,91E 11	4,66E 08	4,84E 10	1	0	0		Gene Matrix
f13	1,54E 09	4,06E 09	2,56E 09	1,40E 08	1	0	1	Gene Matrix	Gene Matrix
f14	6,36E 12	2,97E 11	2,66E 12	6,42E 12	1	0	0		Canonical
f15	4,65E 01	2,23E 01	6,20E 01	3,53E 01	0	1	1	Gene Matrix	Gene Matrix
f16	2,16E+84	8,23E+84	1,55E+85	2,72E+85	1	1	1	Gene Matrix	Gene Matrix
f17	6,45E 05	4,47E 05	3,84E 05	2,85E 05	1	1	1	Canonical	Canonical
f18	1,10E 02	3,11E 02	1,66E 02	3,68E 02	1	0	1	Gene Matrix	Gene Matrix
f19	1,68E+01	5,52E+00	2,17E+01	6,10E+00	0	1	1	Gene Matrix	Gene Matrix
f20	4,44E+01	4,28E+01	2,77E+01	3,99E+01	1	1	1	Canonical	Canonical
f21	5,97E+02	3,76E+02	1,65E+03	3,74E+02	0	1	1	Gene Matrix	Gene Matrix
f22	4,58E+00	3,81E+00	3,43E+00	3,65E+00	1	0	0		Canonical
f23	1,87E 08	7,05E 08	3,18E 09	6,60E 09	1	0	0		Canonical
f24	1,10E 09	2,92E 09	2,85E 10	8,25E 10	1	1	1	Canonical	Canonical
f25	1,01E 07	3,40E 07	7,44E 09	1,63E 08	1	1	1	Canonical	Canonical
f26	5,30E 06	6,63E 06	7,94E 06	3,63E 05	1	0	1	Gene Matrix	Gene Matrix
f27	1,97E 03	1,33E 02	9,30E 04	6,47E 03	1	0	1	Canonical	Canonical

Table 4.11: Test-set functions overview and results for population size thirty

id	Gene Matrix results		Canonical results		Statistical tests results			Techniques comparison	
	std	mean	std	mean	normal	ttest	wilcoxon	statistical best	best
f1	8,76E 06	1,08E 05	2,55E 06	2,30E 06	1	1	1	Canonical	Canonical
f2	3,50E 13	1,07E 12	3,05E 02	1,51E 01	1	0	0		Gene Matrix
f3	1,83E 11	6,26E 11	1,09E 12	4,08E 12	1	0	1	Canonical	Canonical
f4	3,22E 13	1,34E 12	1,73E 12	8,28E 12	1	0	0		Gene Matrix
f5	3,58E 07	3,52E 12	3,58E 07	2,87E 10	1	0	0		Gene Matrix
f6	1,58E 01	1,11E+00	9,30E 02	6,58E 01	1	0	1	Canonical	Canonical
f7	8,71E 01	9,53E 01	1,09E+00	9,66E 01	1	0	0		Gene Matrix
f8	2,80E 01	4,54E 01	1,16E 01	3,01E 01	1	1	0		Canonical
f9	4,09E 14	1,02E 13	4,58E 12	2,42E 11	1	0	1	Gene Matrix	Gene Matrix
f10	1,89E 02	2,34E 02	1,85E 02	2,49E 02	0	0	0		Canonical
f11	5,01E 02	5,94E 02	3,34E 02	5,41E 02	0	0	1	Canonical	Canonical
f12	4,65E 08	7,41E 13	4,65E 08	2,97E 11	1	0	0		Gene Matrix
f13	1,67E 11	5,14E 11	2,17E 12	4,34E 12	1	1	1	Canonical	Canonical
f14	2,44E 14	7,51E 14	3,94E 14	1,15E 13	1	0	0		Gene Matrix
f15	3,56E 01	2,04E 01	4,76E 01	3,42E 01	0	1	1	Gene Matrix	Gene Matrix
f16	1,79E+84	7,59E+84	5,74E+84	2,54E+85	1	0	0		Gene Matrix
f17	2,22E 05	2,46E 05	1,55E 05	1,83E 05	1	1	1	Canonical	Canonical
f18	7,49E 03	2,25E 02	6,23E 03	1,46E 02	1	0	0		Canonical
f19	1,56E+01	6,46E+00	1,64E+01	4,88E+00	0	0	0		Gene Matrix
f20	1,78E+01	3,43E+01	1,89E+01	3,17E+01	1	0	0		Gene Matrix
f21	8,52E+02	5,37E+02	1,55E+03	3,41E+02	0	1	1	Gene Matrix	Gene Matrix
f22	3,69E+00	3,76E+00	3,40E+00	3,75E+00	1	0	0		Canonical
f23	3,83E 10	1,47E 09	1,69E 09	1,10E 08	1	0	0		Gene Matrix
f24	6,22E 12	1,27E 11	1,17E 12	2,55E 12	1	1	1	Canonical	Canonical
f25	6,52E 10	2,91E 09	5,94E 11	1,19E 10	1	0	1	Canonical	Canonical
f26	4,71E 08	7,05E 08	1,70E 08	5,20E 08	1	1	1	Canonical	Canonical
f27	1,30E 04	9,00E 04	4,61E 05	3,26E 04	1	0	1	Canonical	Canonical

4.4 Conclusions

The Evolutionary Strategy Learned with Automated Termination Criteria (ESLAT) technique introduced some interesting tools to control the population diversity (gene matrix and mutagenesis operator) and establish, according to it, a stopping criterion. However, several implementation handicaps were pointed out and remained unsolved: control over the search domain, local search techniques configuration, robustness in the stopping criterion, etc.

Robust ESLAT (R-ESLAT) faces these issues, providing an analysis of the exploration and exploitation processes carried out by the technique, along with concrete measures involving each of them. This measures can be grouped in three main fronts: search space control, technique configuration (particularly for the intensification processes) and robustness in the stopping criterion included. Search space control has been performed with both repairing and death penalty approaches. Intensification processes have been revised setting appropriate configuration values for the local search techniques, along with complementary required techniques.

Finally, an exploitation analysis (by means of the best individual's fitness value) has been included in the stopping criterion. This stopping criterion now combines a stopping prevention mechanism (by the diversity introduction guided by the gene matrix and mutagenesis procedures) with the combined stopping detection (both in objective and variable spaces), providing a much more robust assessment of stopping situations. The presented algorithm produces results which are statistically better than CMAES in terms of final quality in 20 out of the 27 functions in the used dataset, showing more consistency in the results through the different executions, and also overcoming the difficulties of the original ESLAT algorithm.

These results showed promising capabilities, but were intrinsically included as part of the memetic algorithm cycle. The developed mutagenesis based stopping criterion isolates these gene matrix and mutagenesis artifacts and focuses on their diversity enhancement, redefining the processes in order to maximize these characteristics, and tests the results comparing them to the performance of canonical evolution strategies. This implies that the passive role of a stopping criterion is change to a more active stopping prevention and detection, according to measures both in the variable and the objective spaces.

The obtained results show that the exploration improvements lead the algorithm to an overall better performance, with a different impact regarding the population size and the percentage of the population which goes through mutagenesis processing. For a set of twenty-seven unconstrained optimization functions, the algorithm is statistically better considering four different population sizes and fifty iterations, providing a fair statistical significance. The testing process also highlights the requirement for mean centered statistical tests, since non-parametric alternatives may not be able to measure performance differences under certain specific circumstances due to their median analysis.

The resultant performance provides higher exploration capabilities and resistance to the effect of local optima. This computational cost is substracted from the exploitation processes, which makes the algorithm obtain worse final results than canonical techniques when their optimization is successful. This creates results which, for some problems, as already commented, are statistically not comparable.

As established in the thesis objectives, this chapter has presented a novel proposal for a single objective stopping criterion. This criterion measures stagnation in both variable and objective space before it is triggered, and actively manipulates the population diversity in the

variable space to prevent early convergence issues.

5

Multi-objective stopping criteria for evolutionary algorithms

> *Delirium was once Delight And although that was long ago now, even today her eyes are badly matched; one eye is a vivid emerald green, spattered with silver flecks that move; her other eye is vein blue Who knows what Delirium sees, through her mismatched eyes?*
>
> Neil Gaiman, *Sandman-Season of Mists*, 1992

This chapter presents a proposal to approach the stopping criteria for multi-objective evolutionary algorithms. The core idea of this chapter is to use quality indicators from quality assessment literature (Zitzler et al., 2003) to measure the progress of the algorithm across different generations and determine, according to them, determine when the algorithm must be stopped. The first proposal, presented in section 5.2, will include a Kalman estimation technique to handle the linearity of the indicator values and analyze the inclusion of possible fusion architectures in order to manage several indicators jointly and add robustness to the final stopping criterion. The second proposal, presented in section 5.3, introduces the Least Squares Stopping Criterion, LSSC, which is focused, according to previous results, on a simplified approach for the linear estimation using least squares and attempting to provide an iteratively computed criterion which can be implemented as a single formula, in order to enhance its simplicity and enhance its inclusion in available algorithms. The main reference texts for this chapter are (Guerrero et al., 2009a, 2010c).

5.1 Introduction

Most soft computing methods (both heuristic and non-heuristic) share the need for a stopping criterion in their design. This need is usually met applying criteria based on the number of generations, which involve a waste of computational resources (as they go on being applied after a point where iterations get no improvement over the current solution). According to the classifications presented in section 2.4.7, these approaches are static, exhaustion-based criteria. Even though the waste of resources is, evidently, never a desirable attribute, it becomes especially important in real systems where the running time becomes a critical

parameter. This fact has prevented the application of some of those algorithms to real problems, due to the time needs required.

Multi-objective optimization problems (MOOP's) (Ehrgott, 2005) are optimization problems where a group of functions, usually in conflict, has to be optimized jointly. They were presented and properly formulated in section 2.4.3, as a generalization over the single objective optimization goal of evolutionary algorithms. The solution to this problem is a set, known as Pareto Optimal Front (definition 2.5.4 and equation 2.7), which contains one or more feasible solutions corresponding to the extreme values (either maximum or minimum, depending on each particular case) of the functions. As previously discussed, these definitions rely on the concept of Pareto dominance (definition 2.5.2) and Pareto optimality (definition 2.5.1).

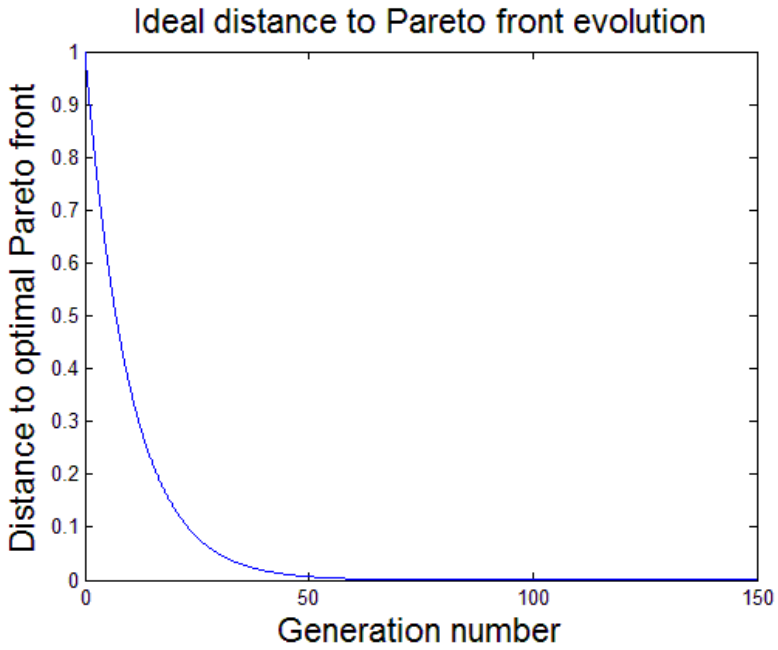
Multi-objective evolutionary algorithms (MOEA's) (Coello et al., 2007) have proved to be a particularly useful tool to deal with MOOPs (covered in section 2.5). In this particular field, the need for a stopping criterion, even though it is sometimes left aside as a secondary matter, is highlighted in surveys as a topic which needs to get attention and research in this area (Coello, 2000), being a recent research focus (Wagner et al., 2009). The traditional solution to this problem was the establishment of an a priori number of maximum generations for the algorithm (Coello, 2000), which, at least, managed to set a boundary for the algorithm's running time. This need for a stopping criterion has already been stated and approached in general for EAs (Hernandez et al., 2005; Safe et al., 2004). Chapter 4.3 has already dealt with this topic, presenting a novel stopping criterion for the single objective case.

Being the most commonly applied criterion, in any of the problems exposed, to stop the running algorithm after it reaches a number of iterations (usually measured in generations) it is remarkable that this stopping criterion can only be applied to very concrete and simple MOOP's, becoming unfeasible in more complex problems. Setting that a-priori value can be a particularly difficult task to be performed accurately, since this issue is a MOOP itself, where the objectives are to maximize the quality of the optimal Pareto front (OPF) approximation while minimizing the number of generations, or, in general, the number of function evaluations. This fact can be somehow avoided in some studies considering that the number of required iterations is rather established for well-known problem suites, such as DTLZ (Deb et al., 2002c) or WFG (Huband et al., 2005).

We may use a local or a global approach to find a stopping criterion for our EA. A local criterion (or iteration-wise) only has access to the current iteration's data, measuring the distance from this value to the predefined optimum, and stopping after this distance gets lower than a certain threshold. This method requires us to know the optimal solution before applying the algorithm, so it may be only used for validation purposes. Global criteria, on the other hand, keep track of the advances obtained by the algorithm over several iterations, in order to take the decision of whether to stop or not. The distance measure definition is still needed in these criteria, but not the knowledge of the optimal solution.

The stopping criterion of a MOEA is typically invoked at the end of an iteration of the algorithm, deciding whether to continue with the next generation or not. There are four situations where this stop should take place: the solution is already satisfactory, a better solution is unlikely to be produced, the method is unable to converge to a solution or, finally, the amount of computation is already sufficient. The first situation is covered by local approaches. The second and third situations are covered by global approaches, by the comparison of consecutive iterations. The fourth situation is covered by stopping criteria

Figure 5.1: Ideal distance evolution from a MOEA solution to the Optimal Pareto Front



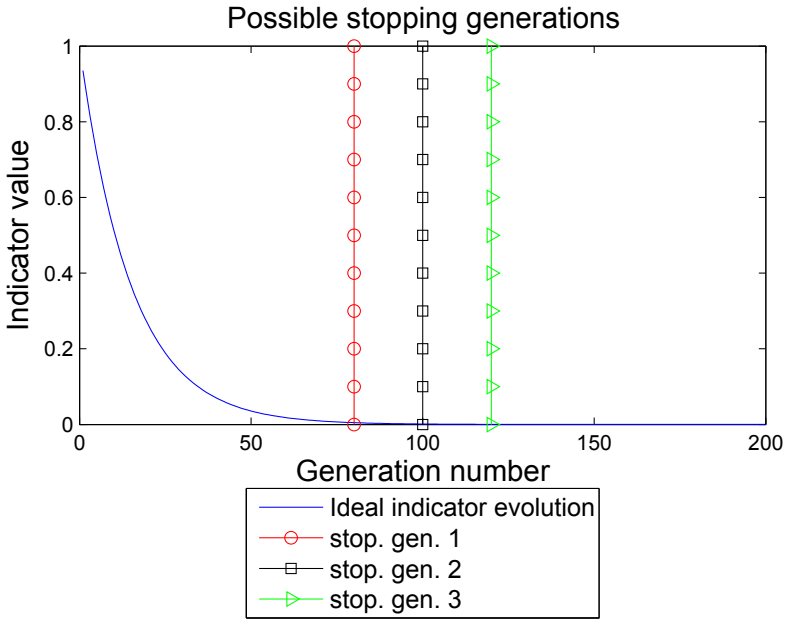
triggered after a certain predefined number of generations.

There are examples of these criteria applied to specific algorithms, such as NSGA-II (Deb et al., 2002b), where a stopping criterion was proposed based on some algorithm internal measures, such as the crowding distance, (Roudenko & Schoenauer, 2004).An important related concept is the *ideal stopping generation*. Stopping generation criterion is a MOOP itself, where the stopping of the evolutionary process should be triggered as soon as possible (once this process is not improving its solution), being this solution, at the same time, as close to the optimal Pareto front as possible. We can understand our concept of ideal stopping generation with Figure 5.1 is introduced to ease the definition of this *ideal stopping generation*.

In figure 5.1 we may see that the progress made towards the solution ahead of generation fifty is very small, so that would be our ideal stopping generation. The balance between the quality of the solution of a MOEA and the number of the generations it has been running for can be found, as has been exposed, with the use of global stopping criteria, but there are, as in any MOOP, some decisions which the user must take in order to determine which solution he wants to use (stating it simply, which grade of progress towards the solution per generation is worth to keep the algorithm running). It must be noted, as well, that the traditional approach is not always better, in terms of solution quality, as additional generations may degrade factors such as the population diversity by means of genetic drift (Rudolph et al., 2007). This implies that, by choosing the right stopping generation, the user is not only saving computational resources, but may be also preserving the quality of the obtained solution.

Figure 5.2 shows three possible stopping generations for an ideal evolution results. There is no clear choice regarding the best stopping generation in Figure 1. In a general case (the one which the default configuration of a stopping criterion should be trying to achieve) the

Figure 5.2: Different possible stopping generations for a given indicator evolution



optimal stopping generation would be number two, but there may be additional considerations. If the running time of each individual generation is very high, the user might try to stop the algorithm following stopping generation number one (after which, even if there is some improvement in the solution quality, it is very low), whereas, on the other hand, if extreme accuracy in the solution is the key factor in the algorithm, the user might prefer stopping generation three. This shows that stopping criteria must be able to adapt to the user's needs.

The required distance functions are an issue themselves. In section 2.6 the techniques required for quality assessment of the performance of different MOEAs were introduced, along with the requirements of quality indicators and the criticism over some of the most extended ones (particularly unary quality indicators). Three different quality indicators will be presented and used in this chapter: hypervolume indicator (Zitzler & Thiele, 1998), epsilon indicator (Zitzler et al., 2003) and mutual domination rate indicator (MDR) (Martí et al., 2007). A capital difference among these three alternatives is that MDR was especially designed as a convergence indicator (part of the MGBM stopping criterion introduced in section 2.7.2), while the other alternatives have been adapted from an initial quality assessment purpose.

Accumulating evidence by means of linear estimation from quality indicators is exposed in (Martí et al., 2007, 2009), but without the proper study of the limitations imposed by the chosen filtering. Two different alternatives will be presented to cope with this process, a novel use of Kalman filtering and an approach based on least squares regression. Finally, the idea of using the proposed stopping criterion as a part of a data fusion architecture is presented, using the previous indicator's data as the input data for the established stopping criterion. A data fusion architecture aims to combine data from multiple sources to determine the state of a system (Groves, 2008). This architecture will take the evidence accumulator role. In this chapter we will propose a Kalman estimation based stopping criterion, from several progress indicators, and introduce of fusion architectures to consider the indicators' data jointly.

5.2 Linear estimation based stopping criteria for MOEAs approach

5.2.1 Progress Indicators

As presented in the introductory section of this chapter, three quality indicators will be used for the data gathering of the proposed stopping criteria: hypervolume (Zitzler & Thiele, 1998), epsilon (Zitzler et al., 2003) and MDR (Martí et al., 2007) indicators. The nomenclature *progress* indicator refers to the use of the indicator, and the way it is calculated, in this case to measure the convergence of the given multi-objective algorithm, instead of the traditional quality assessment presented in section 2.6. Even though this fact will be thoroughly discussed in the following sections, this implies that the binary quality indicators will measure the progress obtained between two different instants of the evolutionary cycle (two different generations). This must be taken into account to perform the proper modification over hypervolume and epsilon indicators to suit this goal, while MDR was inherently designed to cover this approach.

Measuring the quality of solutions

Since our progress indicators are going to be derived from quality ones, it is therefore important to realize what these indicators measure and how they do it. It is a complex and crucial matter to be able to determine how good is a solution space in relationship to the optimal one, involving problems such as dimensionality reduction, which may lead to invalid conclusions (Zitzler et al., 2002). However there are some community accepted indicators (Zitzler et al., 2003), which can be grouped in three categories: distance from the Pareto front solution's elements to the closest from the optimal one (to measure how close our solution is to the optima), distance from every element of the optimal Pareto front to the closest element of the actual solution (to determine how well our solution covers the optimal one) and finally the distribution of the actual solution and its associated Pareto front (to gauge how well spread are the elements of these sets). Binary indicators (Zitzler et al., 2003) are especially indicated for our purpose, since they compare two different sets of solutions.

Hypervolume indicator

The hypervolume indicator $I_h(A)$ (Zitzler et al., 2007) computes the volume of the region H, delimited by a given set of points A, and a set of reference or nadir points N. Equation 5.1 presents the unary version of the indicator (Zitzler & Thiele, 1998), while equation 5.2 presents the binary version (Zitzler & Künzli, 2004).

$$I_h(A) = volume\left(\bigcup_{\forall z \in A; \forall y \in N} hypercube(a, n)\right) \tag{5.1}$$

$$I_{HD}(A, B) = \begin{cases} I_H(B) - I_H(A) & \text{if } \forall x^2 \in B \exists x^1 \in A : x^1 \succ x^2 \\ I_H(A + B) - I_H(A) & \text{in any other case} \end{cases} \tag{5.2}$$

It must be noted that not only are there different approaches for the hypervolume indicator (probably the most extended quality indicator) in terms of the number of sets of solutions

used, but also regarding the way the computation is performed. The implementations are generally based on the *hypervolume by slicing objectives* approach (Knowles, 2002; While et al., 2006). This indicator has gone under an incremented use by the development of indicator based algorithms IBEAs (Zitzler & Künzli, 2004), which include indicators computation as part of the evolutionary cycle. The difficulties with this approach are particularly evident in algorithms dealing with many-objectives, such as HypE (Bader & Zitzler, 2011), due to the exponential worst-case runtime complexity of the indicator in the number of objectives, more specifically $\mathcal{O}(N^{n-1})$, where N is the number of solutions considered (Knowles, 2002).

Nadir points are the worst elements of our solution's Pareto front, which means the points which dominate no other. The computation of these nadir points determines the accuracy of the indicator. Having N, the computation of this indicator is also a non-trivial, computationally intense problem (Ehrgott & Tenfelde-Podehl, 2003).

Epsilon indicator

The epsilon indicators (Zitzler et al., 2003) are a set of performance indicators which, relying on the dominance concept, measure how close our actual Pareto front is to the global optimal one. They introduced the epsilon dominance concept (presented in section 2.6.1, covered along with the most important remaining dominance relations in definition 2.6.1 and compared in tables 2.1 and 2.2).

Suppose without loss of generality a minimization problem with n positive objectives, i.e. $Z \subseteq \mathbb{R}^{+^n}$. An objective vector $\vec{z}^1 = (z_1^1, z_2^1, \dots, x_n^1)$ is said to ϵ-dominate another objective vector $\vec{z}^2 = (z_1^2, z_2^2, \dots, z_n^2) \in Z$, written as $z^1 \succeq_\epsilon z^2$ if, and only if

$$\forall 1 \leq i \leq n : z_i^1 \leq \epsilon \bullet z_i^2 \tag{5.3}$$

for a given $\epsilon > 0$. The binary epsilon indicator I_ϵ is therefore defined as

$$I_\epsilon(A, B) = \underset{\epsilon \in \mathbb{R}}{inf} \{\forall \vec{z}^2 \in B \exists z^1 \in A : z^1 \succeq_\epsilon z^2\} \tag{5.4}$$

Similarly, an additive epsilon indicator can be defined: $z^1 \succeq_\epsilon z^2$ if, and only if

$$\forall 1 \leq i \leq n : z_i^1 \leq \epsilon + z_i^2 \tag{5.5}$$

for a given $\epsilon > 0$. The binary additive epsilon indicator $I_{\epsilon+}$ is therefore defined as

$$I_{\epsilon+}(A, B) = \underset{\epsilon \in \mathbb{R}}{inf} \{\forall \vec{z}^2 \in B \exists z^1 \in A : z^1 \succeq_{\epsilon+} z^2\} \tag{5.6}$$

Mutual Dominance Rate Indicator

The two previous indicators, as has been previously pointed out, were formulated as quality indicators and have to be reformulated to address the convergence issue. The main handicap of such an approach is the fact that for offline quality assessment the computational efficiency is, in general, not a key parameter (the computational complexity of the hypervolume indicator has just been related in its section to the issues appearing in many-objectives optimization), and thus their application as an step of every generational cycle of an evolutionary algorithm may seriously harm the overall performance.

The MDR indicator (Martí et al., 2007) is a specially created indicator to deal with this computational cost weakness. To simplify the indicator definition, a function $\triangle(A, B)$ that returns the set of elements of A that are dominated by at least one element in B, as shown in equation 5.7.

$$C = \triangle(A, B) \text{ such that } \forall x \in C, x \in A \text{ and } \exists y \in B \text{ with } y < x \tag{5.7}$$

Using the function defined in equation 5.7, the MDR indicator, $I_{mdr}(t) \in [-1, 1]$ contrasts how many non-dominated individuals of iteration t dominate the non-dominated individuals of iteration *t-1* and viceversa, using equation 5.8

$$I_{mdr}(P^*_t, P^*_{t-1}) = \frac{\| \triangle (P^*[t-1], P^*[t])\|}{\|P^*[t-1]\|} - \frac{\triangle(P^*[t], P^*[t-1]\|}{\|P^*[t]\|} \tag{5.8}$$

If $I_{mdr} = 1$, it means that the population of iteration t is completely better than the precedent one. If its value is 0, it implies that there has not been any substantial progress. If $I_{mdr} = -1$, it indicates the worst possible case, where the iteration t does not improve any of its predecessor's solutions. The formulation of the MDR indicator was presented along with a stopping criteria named MGBM (Martí et al., 2009), which was discussed in section 2.7.2. This implies that even in its formulation, it is presented to compare two different Pareto fronts of the same evolutionary algorithm, instead of two different sets of solutions to be compared.

We may introduce the same formulation to the hypervolume and epsilon indicators previously presented, in order to use them as progress indicators, instead of tools for quality assessment. From equation 5.2 we obtain the following binary hypervolume indicator for progress assessment.

$$I_{HD}(PF^*[t-1], PF^*[t]) = \begin{cases} I_H(PF^*[t]) - I_H(PF^*[t-1]) \text{ if } \forall x^2 \in B \exists x^1 \in A : x^1 \succ x^2 \\ I_H(PF^*[t-1] + PF^*[t]) - I_H(PF^*[t]) \text{ in any other case} \end{cases} \tag{5.9}$$

Both formulations of the epsilon indicator, multiplicative (equation 5.4) and additive (equation 5.6), can also be reformulated to meet the requirements of a progress indicator, with equations 5.10 and 5.11 respectively.

$$I_\epsilon(PF^*[t-1], PF^*[t]) = \inf_{\epsilon \in \mathbb{R}} \{\forall \vec{z}^2 \in PF^*[t] \exists z^1 \in PF^*[t-1] : z^1 \succeq_\epsilon z^2\} \tag{5.10}$$

$$I_{\epsilon+}(PF^*[t-1], PF^*[t]) = \inf_{\epsilon \in \mathbb{R}} \{\forall \vec{z}^2 \in PF^*[t] \exists z^1 \in PF^*[t-1] : z^1 \succeq_{\epsilon+} z^2\} \tag{5.11}$$

5.2.2 Kalman Linear Estimation

The Kalman filter was originally presented in (Kalman et al., 1960) as a recursive solution to the discrete data filtering problem, being ever since a subject of extensive research and application, particularly in the area of autonomous or assisted navigation. The Kalman filter

assumes a dynamic linear model and measurement process, both with added white Gaussian noise, given by the following equations.

$$X[k] = A[k-1]X[k-1] + w[k-1] \tag{5.12}$$

$$Z[k] = H[k]X[k] + v[k] \tag{5.13}$$

For equations 5.12 and 5.13 $w[k] \sim N(0, Q)$ and $v[k] \sim N(0, R)$
The filter works along with an estimated error covariance matrix, given by equation 5.14

$$P_K[k] = A[k-1] * PK[k-1] * A[k-1]^t + Q[k-1] \tag{5.14}$$

The relationship between the used indicators and the models represented by a Kalman filter is not immediate. Current proposal will use a model tracking our indicator's value and its first derivative in the state vector (being our only measure the indicator value). The following values for the filter parameters are delimited by this choice

$$X[k] = \begin{bmatrix} x[k] \\ x[k] \end{bmatrix} \qquad Z[k] = \begin{bmatrix} z[k] \end{bmatrix} \qquad H[k] = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.15}$$

The $w[k]$ noise factor (along with its covariance matrix Q) represents the uncertainty in our model, which allows us to adapt to dynamically changing linear models. The $v[k]$ noise factor (along with its covariance matrix R) represents the noise in our measuring process. In the current proposal, where the R matrix only has one element, it represents the variance in our measures.

$$Q = \begin{bmatrix} \sigma_x^2 & \sigma_{xx}^2 \\ \sigma_{xx}^2 & \sigma_x^2 \end{bmatrix} \qquad R = \begin{bmatrix} \sigma_Z^2 \end{bmatrix} \tag{5.16}$$

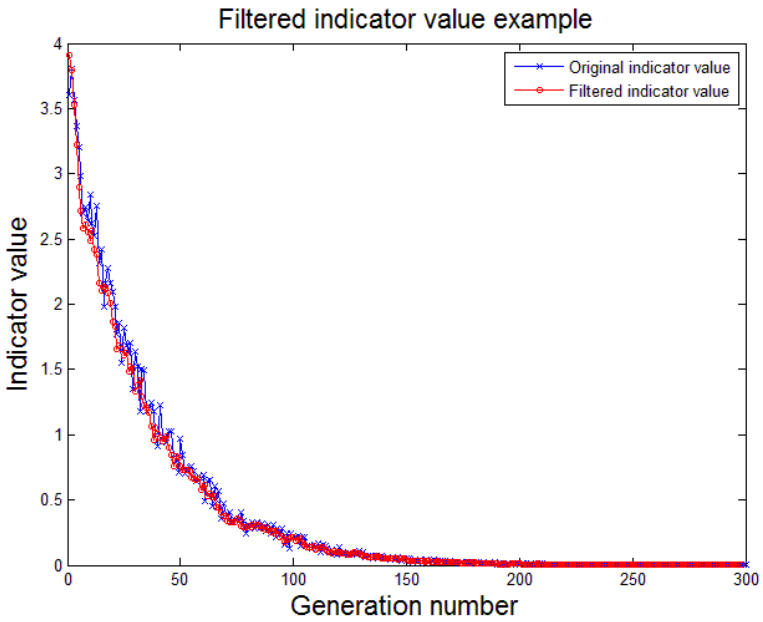
Relating these values to the stopping situations presented in section 5.2.1, we can easily associate the state vector with the situations where a stop is required due to the unlikeness of getting a better solution or converging to one.

It is important to realize the strong suppositions that are taken by the use of a Kalman filter with the indicated model. It assumes a linear model with Gaussian noise with a constant variance (just by using a Kalman filter) and constant velocity in the change rate of the indicator (due to our chosen model). These preliminary conditions are not satisfied (especially during the transient evolution of the indicators), but we will use them anyway as an approximation to our problem, given that the transient phase of the evolution is not of interest to the stopping criterion.

It is important that, even though we do not know the model for the transient state (and it does not follow our Kalman filter suppositions) we do know that it will start to follow an almost uniform model at the end of the transient state, which is precisely when we want to trigger our stopping criterion. The commented effect can be appreciated in figure 5.3.

We have already shown the P_K matrix as an estimation of the filtered error (both for the state vector position and the velocity). The first statistical stopping criterion would determine when the generations have stopped advancing towards the desired solution as a function of current indicator derivative and the estimated error for that derivative. Unfortunately, due to the approximation limitations, we have a very pessimistic error prediction, implying that we

Figure 5.3: Filtered indicator example



cannot use a strictly statistical criterion to determine our stopping generation (the estimated error is greater than the measure).

Another possibility is to determine the stopping generation by tracking the residue value, being that residue the difference between our predicted point and the real one (in this case, the output of the Kalman filter versus the measured indicator). This residue is shown in equation 5.17

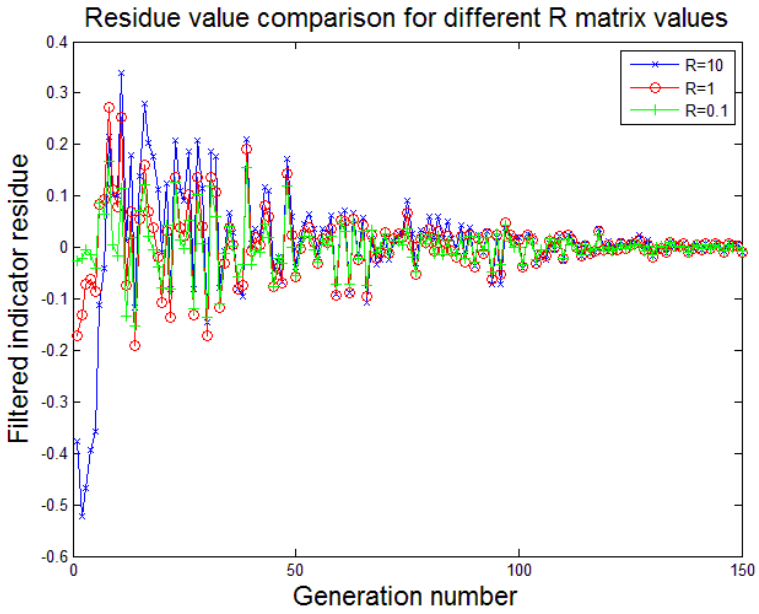
$$Res[k] = |\hat{x}[k|k-1] - z[k]| \tag{5.17}$$

This residue study, applied to the aircraft position tracking with a Best Linear Unbiased Estimator (BLUE) (Henderson, 1975), was successfully applied in chapter 3 to the segmentation of data coming from the ATC domain. The problem that arises for current application is that, during the update phase of the Kalman filter, we adapt our prediction to the measured value, according to the value of R, so the lower the R value is, the smaller the residue will be. This fact is shown in figure 5.4

This means that, by tuning R and Q matrices, we would be able (as we already know the optimal Pareto front for our domain and therefore the optimal stopping generation) to trigger our stopping criterion at optimal generation for each of our problems individually, but these results would not be applicable to different ones, as we would be basing ourselves on local criteria, which have been covered on previous sections.

We have stated the approximations made to be able to apply Kalman filtering to the included indicators (which follow a non-linear model). It is especially important the difference between the transient state, where the model does not follow the real evolution of the indicator, from the final steady state (at the beginning of which we would like to stop), where this model's conditions are met. If we choose a high value of R matrix (meaning that we do not trust much our measures value) and a low value of Q matrix (meaning that we do

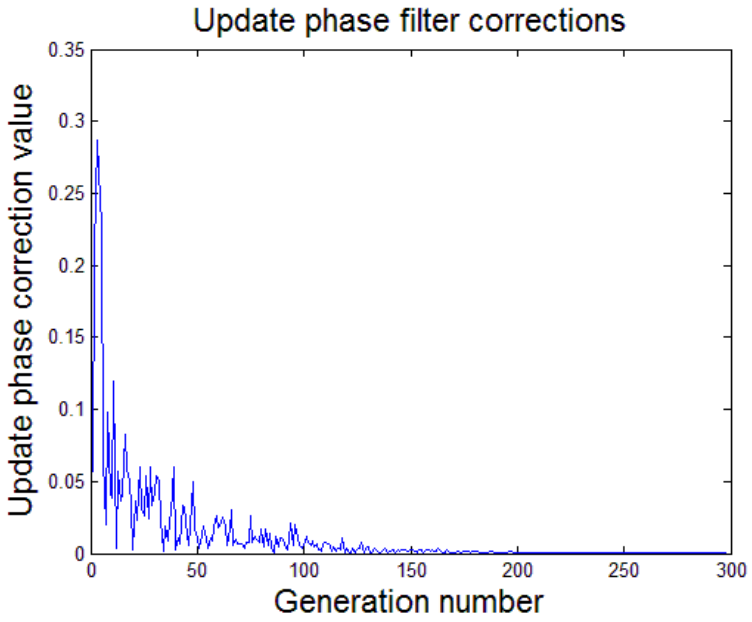
Figure 5.4: Filtered indicator residues comparison



trust our model) we would get a filter adapting itself awfully to the first transient state, but whose results during out final stopping phase should become much better. We may present this as the difference between our filter's prediction phase and the filter's update phase. The lower this value gets, the more accurate the filtering becomes, thus the more the prediction is adapting itself to the measures and finally the closer the MOEA is to its stopping generation. Figure 5.5 shows the evolution of these update phase corrections for Q=0.01 and R=10. We have stated the approximations made to be able to apply Kalman filtering to the included indicators (which follow a non-linear model). It is especially important the difference between the transient state, where the model does not follow the real evolution of the indicator, from the final steady state (at the beginning of which we would like to stop), where this model's conditions are met. If we choose a high value of R matrix (meaning that we do not trust much our measures value) and a low value of Q matrix (meaning that we do trust our model) we would get a filter adapting itself awfully to the first transient state, but whose results during out final stopping phase should become much better. We may present this as the difference between our filter's prediction phase and the filter's update phase. The lower this value gets, the more accurate the filtering becomes, thus the more the prediction is adapting itself to the measures and finally the closer the MOEA is to its stopping generation. Figure 5.5 shows the evolution of these update phase corrections for Q=0.01 and R=10.

We have as designers two degrees of freedom: one related to the threshold below which we would consider we have reached our stopping generation, and another one related to how many measures we will require to get consecutively with values below the threshold to actually make the MOEA stop. The first parameter should be indicator dependent but problem independent, and, along with the second one, is a compromise between the cost of additional iterations and the cost of stopping before we reach the optimal generation. The number of measures also adds robustness to our criterion, since, as we can see in the figures included in the experimental section (figures 5.14-5.18), we may have measures during the

Figure 5.5: epsilon indicator update phase corrections with Q=0.01 and R=10



transient indicator state that fall below our threshold value. These measures can be used as well to try to escape from local minimal solutions, and give the MOEA the chance to keep looking for a better solution.

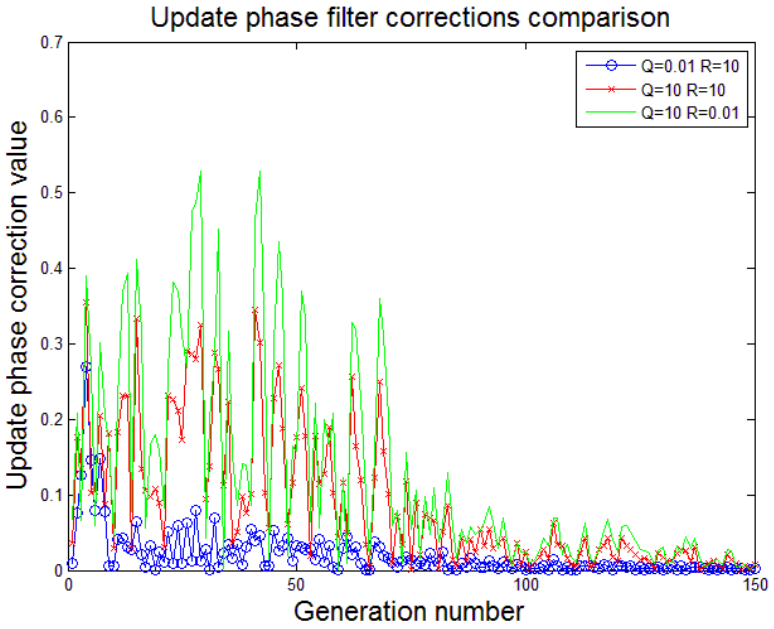
The determination of these parameters may be seen as parameter optimization problem itself, since we have to choose the best filter parameters to allow the stopping criterion to get a distribution of the update corrections where we can choose an appropriate threshold and number of measures. Figures 5.6 and 5.7 show comparative values to remark the importance of these parameters.

In figures 5.6 and 5.7, it can be seen that the value for the threshold depends heavily on the filter parameters used. The corrections made during the update phase depend directly on the value of the R matrix (the more we trust our measures, the more we will correct our prediction), but there is not such a clear dependency between the filter parameters, along with the chosen threshold value, with our accuracy determining our stopping generation. In the table beneath we show comparative results for the presented filter parameters. These results come from a DTLZ3 problem (Deb et al., 2002c) using the NSGA-II algorithm (Deb et al., 2002b) (both, problem and algorithm, will be among the dataset used for the experimental validation presented in section 5.2.4).

5.2.3 Indicators combination

Even though the analysis of this comparison is beyond the scope of this chapter, different indicators perform better or worse according to the problem and the algorithm used to solve it. By using one of those indicators alone our stopping criterion would inherit its performance characteristics, which is a situation we would like to avoid if possible. In current proposal, the objective is to combine our different indicators to get a global parameter to determine whether to stop or not. This is performed by data fusion architectures (Groves, 2008).

Figure 5.6: Update phase corrections comparison



There are different alternatives for this fusion, such as a least squares or centralized ones. Data fusion architectures were originally proposed in the context of multisensor information (Liggins et al., 2008), such as combining data from GNSS (global navigation satellite systems, such as GPS) and INS (inertial navigation systems) (Farrell & Barth, 1999).

Least squares data fusion architecture

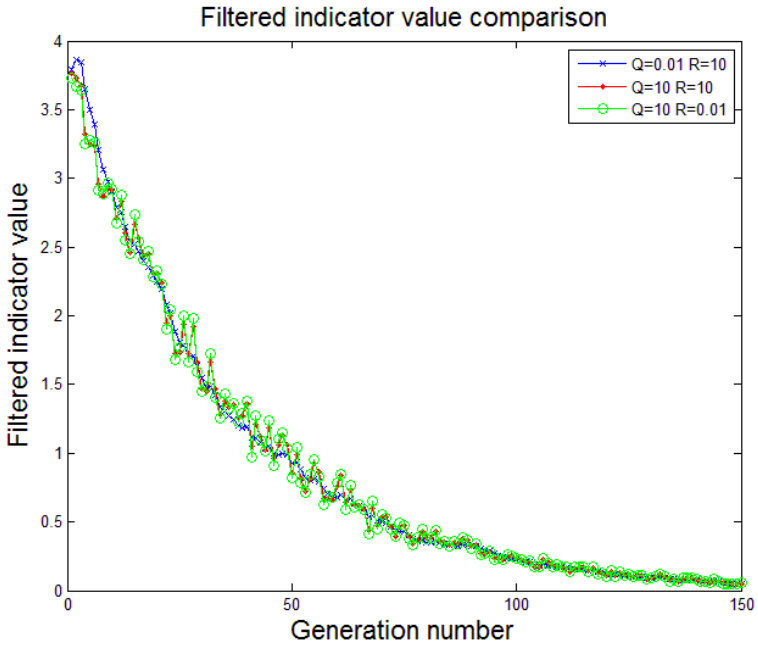
Least-squares integration is the simplest way of combining information from different navigation systems (for current proposal, from different quality indicators). Each system, working as a black box for the integration architecture (meaning that it provides no information about how error varies with time and receives no feedback), gives a position or velocity and position solution ($\hat{x}_i$) and an associated error covariance matrix, P_{ii}. These are combined with a snapshot or single point fusing algorithm (Hegarty, 2006). Figure 5.8 shows an example of this architecture.

The equations for this fusion architecture are quite simpler than those required by alternative ones. Each navigation sensor solution uses different information to obtain its navigation solution (the errors of the different navigation solutions will be uncorrelated), $P_{ij} = 0$ for $i \neq j$. With this simplification we may obtain the final state vector and its associated covariance matrix with equations 5.18 and 5.19

$$\hat{x}_f = P_{ff} \sum_{i=1}^{m} P_{ii}^{-1} \hat{x}_i \tag{5.18}$$

$$P_{ff} = \left(\sum_{i=1}^{m} P_{ii}^{-1} \right)^{-1} \tag{5.19}$$

Figure 5.7: Filtered indicator value comparison



Working with black-box navigation systems creates a need (to be able to optimally combine the different navigation solutions) for very accurate error covariance information. Neglecting the off-diagonal elements of P causes the accuracy to be overestimated in one direction and underestimated in another. Using the off-diagonal elements also allows incomplete navigation solutions to be fused. To do so, the navigation systems must output the information matrix, P^{-1}, instead of the error covariance matrix, as an incomplete navigation solution has infinite uncertainty in one or two directions.

The main advantages of this integration technique are simplicity and low processor load. Also, the fact that all the navigation systems remain independent of each other facilitates integrity monitoring, allowing measurement consistency checks to be used (Brown, 1996). Considering the different quality indicators available in the literature and their different measuring of fronts quality, this consistency checks could be applied in current domain to test whether an individual indicator is failing in its assessment.

Least-squares integration architecture, however, also shows strong limitations. It is not suited for integration of inertial navigation (generally of capital importance to get an accurate short-term integrated navigation solution) or dead-reckoning systems, as it offers no means to calibrate the position drift. In general, for its application to stopping criterion techniques, this should not be a capital handicap, since there is no integrated quality indicator, and thus no drift involved in any of them. Instead of doing this calibration, as these navigation systems' output position drifts, it is weighed out of the integrated navigation position. This fusion architecture also offers no means of combining navigation data with different times of validity, so is unsuited to fast-moving vehicles. Again, for stopping criteria considerations, every progress indicator is measured at the same time (a discrete value represented by the evolutionary generation being measured).

Table 5.1: Stopping generation comparison with different parameter values

Q	R	Threshold	Measures	Stopping generation
0.01	10	0.01	5	101
1	10	0.03	5	104
10	10	0.045	5	106
0.01	1	0.02	5	101
1	1	0.04	5	106
10	1	0.06	5	106
0.01	0.01	0.04	5	106
1	0.01	0.06	5	107
10	0.01	0.07	5	106

Total estate centralized data fusion architecture

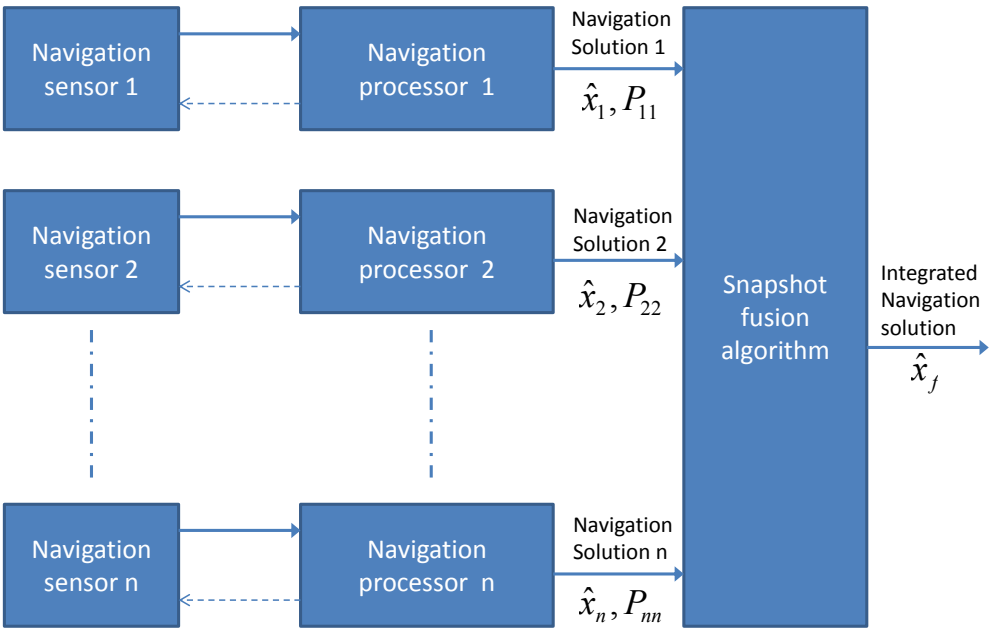
In a centralized integration architecture, sensor measurements (instead of complete navigation solutions) are generally input to the integration Kalman filter. Radio navigation systems provide ranging measurements, enabling a navigation system to contribute to the integrated navigation solution when there are insufficient signals to form its own solution. When the navigation processor does not incorporate any smoothing or estimation algorithm, either the processor or the sensor measurements are acceptable (such is the case of INS, DR or feature matching systems). For current proposal, the raw quality indicator measurements would be the input to the integration Kalman filter. Figure 5.9 shows the overview for this total-state data fusion architecture.

The total-state filter is suited to integrating positioning systems only, whereas an error-state filter is suitable where INS or DR is used. Progress indicators may be considered as positioning systems, so these systems can be considered to be part of a stopping criterion. In a centralized integration architecture, the systematic errors and noise sources of all of the navigation sensors are modeled in the same Kalman filter. This ensures that all error correlations are accounted for, all measurements optimally weighed and the maximum information is used to calibrate each error. Furthermore, we have already pointed out the fact that using cascaded Kalman filters introduced time correlated errors, and thus, using only one Kalman filter for the integration of all the sensor measures (and by that way not introducing the time-correlated errors) we can use higher gains in the filter before we have a stability risk.

According to the previous facts, the centralized integration architecture provides the optimal navigation solution in terms of accuracy and robustness. The main difficulty for this process is having the necessary information to model all sensors correctly (which requires a very careful design). In the case of progress indicators this process is particularly difficult, since there are no clear correlations formally established among them to properly model the required covariance matrices.

Decision fusion architecture proposal

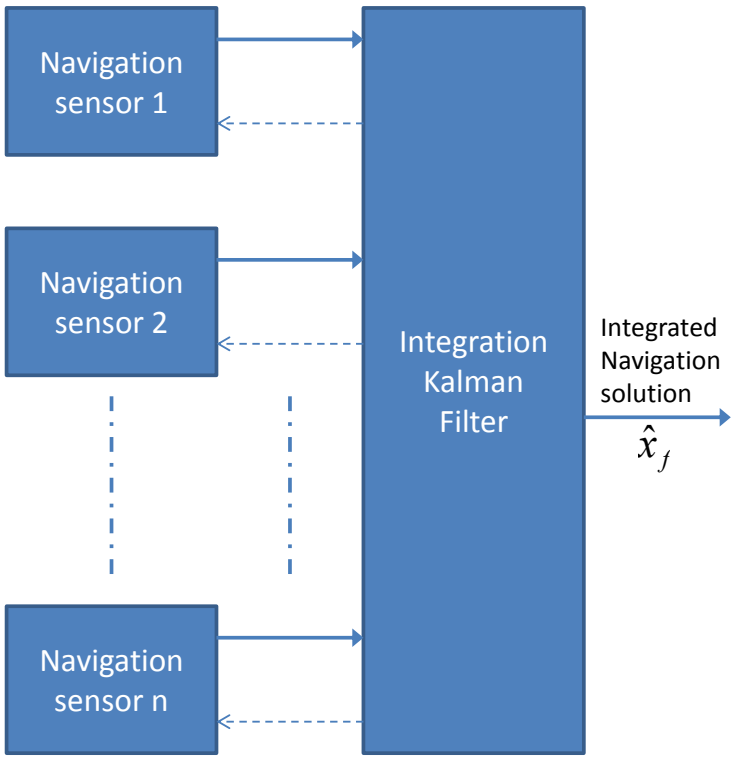
Several issues arise with the application of the architectures presented in section 5.2.3. First of all, least squares integration architectures uses the covariance matrix P_K values as part of its integrated solution procedure, but in section 5.2.2 it has been explained that in our case



these values are higher that their actual value, so alternative architectures which did not rely on such a value so heavily would be preferable.

The case of a total-state integration architecture is slightly more complicated. The basis for the application of the data fusion architecture is the underlying correlation which exists within our indicators (as both of them measure our advance towards the solution), even though we do not know what the exact relation among them is. The application of this fusion architecture would consider the three indicators as measures of a new variable called, for instance, *virtual progress indicator*, that would use the information of the three indicators together. This process requires a careful configuration of the filter matrices and previous data transformations, such as a normalization (the different indicators do not share the same range, so if this pre-normalization is not performed, the changes due to the generations advance towards the optimal Pareto front are masked in those range differences), and the configuration of the covariance matrices according, among other factors, to the relationships among the different indicators. These studies are currently outside of the scope of this chapter.

Assuming a balance between the cost of stopping too early and that of extending our number generations after our optimal stopping one, we will adopt a simple fusion algorithm, which consists on stopping when at least two out of our three stopping algorithms have met their stopping criterion (a simple decision fusion architecture). Keeping this decision fusion architecture, we might choose not to stop until all the indicators have decided so (if the cost of stopping before the optimal stopping generation is greater than that of getting unneeded generations) or as soon as one of them triggers its stopping criterion (if we want to stop as soon as we have a solution available). This decision may seriously affect the robustness of our criterion. For our experimental section we will use a decision fusion based on a voting system (when two have reached their stopping generation, the evolutionary process will be



stopped).This process is shown in figure 5.10.

5.2.4 Experimental validation

Through the previous sections of this chapter, figures detailing indicators' evolutions and the different parameters needed for the proposed stopping criterion have been shown. These results were obtained using and NSGA-II EA along with a DTLZ3 problem. Experimental validation has been based on two different phases: experiments regarding the empirical choice for the different parameters and the test of these parameters with different test cases. This final validation has been performed in two different phases itself: first of all validation with different executions of the same problem and algorithm used to determine the parameters values (section 5.2.4) and finally the test of these of these parameters to a set of different problems and algorithms (section 5.2.4).

Our chosen algorithm set will include: NSGA-II (Deb et al., 2002b), SPEA2 (Zitzler et al., 2001) and PESA(Corne et al., 2000) (the selection of these algorithms is based on their importance to the community and the differences among them). On the problem set we will include DTLZ3, DTLZ6 and DTLZ7 (Deb et al., 2002c), trying to obtain results with a scaling difficulty level to determine how well our stopping criterion adapts to these changes. The details of the configuration of these algorithms and problems can be found in (Martí et al., 2009).

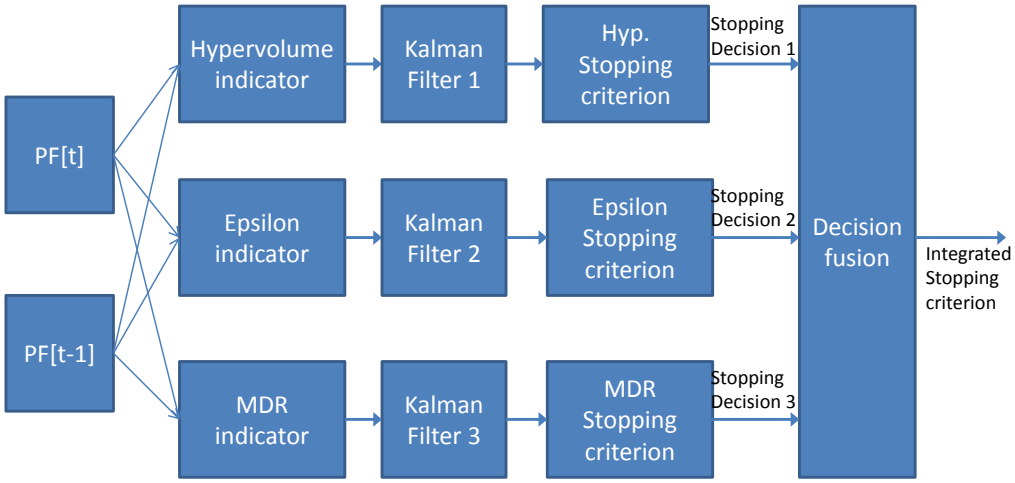


Table 5.2: Experiment parameters to determine the experimental threshold values

Indicator	Q	R	Threshold	Measures	Stopping generation
Hyperv.	1	0.01	0.1	5	93
Epsilon	1	0.01	0.02	5	101
MDR	1	0.01	0.003	5	98

Configuration of the stopping criterion proposal

In the section 5.2.2 the dependency existent within the filter parameters and the chosen threshold value has been presented and explained. Particularly, table 5.1 shows the comparison of how different combinations of the matrices values and the threshold may provide similar results. We proposed values Q=0.01 and R=1 for the filter matrices, being these the base values determining our choice for the threshold. This threshold is indicator dependent as well. Figures 5.11-5.13 show sample evolutions and proposed thresholds corresponding to each of our three indicators for the NSGA-II DTLZ3 problem, to visually justify the choice of the presented experimental thresholds.

Table 5.2 presents the chosen experimental configuration along with the mean stopping generation obtained for the different indicators. A considered matter in this experimental configuration was the similarity between the stopping results of the different indicators, in order to improve the overall robustness of the stopping criterion.

The validation of the results is a matter which must be considered carefully. If we were measuring the success or failure of the MOEA's, we would compute several executions over each problem and algorithm and get the mean distance to the optimal Pareto front. Instead, we want to measure only the success or failure of the proposed stopping criterion. This means that we will have to compare our stopping generation to the evolution of the distance to the Optimal Pareto front, and not just its value at the stopping generation. This distance to the optimal Pareto front has been measured with a hypervolume indicator in its original formulation (as a quality of solution indicator). Figure 5.14 presents, for the problem where

Figure 5.11: Hypervolume indicator sample evolution along with proposed threshold

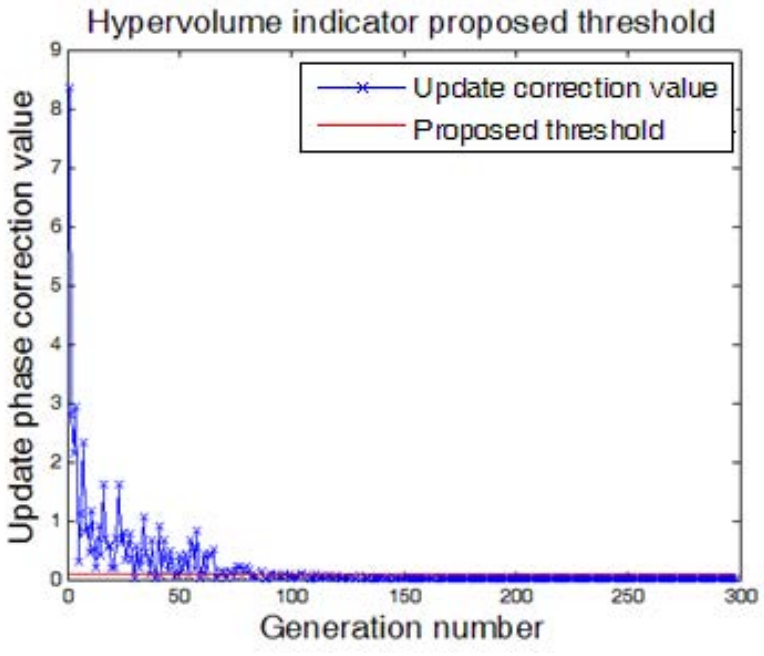


Table 5.3: Stopping generation results for DTLZ3, NSGAII

Indicator	Test number / Stopping generation				
	1	2	3	4	5
Hypervolume	93	108	106	106	107
Epsilon	101	89	91	63	82
MDR	98	101	83	70	95

they were chosen (with the proposed thresholds presented in table 5.2), the distance to the optimal Pareto front and the chosen stopping generations.

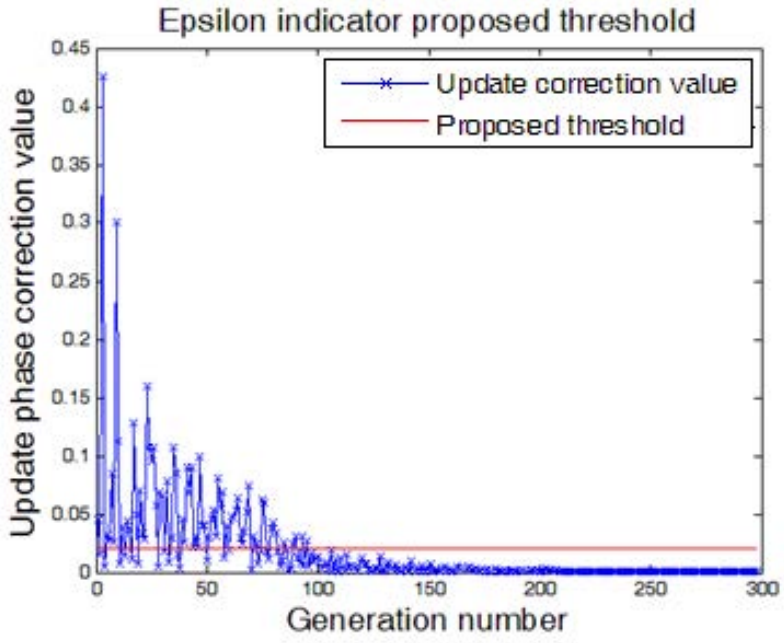
The characteristics proposed in the introductory section of this chapter seem to be met by the results of the stopping behavior presented in figure 5.14.

Validation with different executions

Once we have configured our criterion correctly for one test case, the results obtained for the NSGA-II DTLZ3 problem with different initial conditions will be presented, to test the applicability of the established thresholds to the same problem they were chosen for, but with different initial conditions (and, thus, different evolutions). Figures 5.15 and 5.16 present some of the graphical results for these different evolutions. Also, table 5.3 details the results for five different executions.

Table 5.3 shows that the first execution of the test corresponds to the results presented for the threshold parameter choice (presented in table 5.2). A thorough analysis of the results would require exposing the evolution of the distance towards the optimal Pareto front for each of the shown test cases. Again, there is a lack of knowledge over the optimal

Figure 5.12: Epsilon indicator sample evolution along with proposed threshold



stopping generation, dismissing such an analysis. Figures 5.15 and 5.16 present the results corresponding to the second and third executions of the presented problem, in order to provide a graphical validation test over the proposed stopping criterion, performing an expert validation in the absence of proper statistical testing theory to be applied.

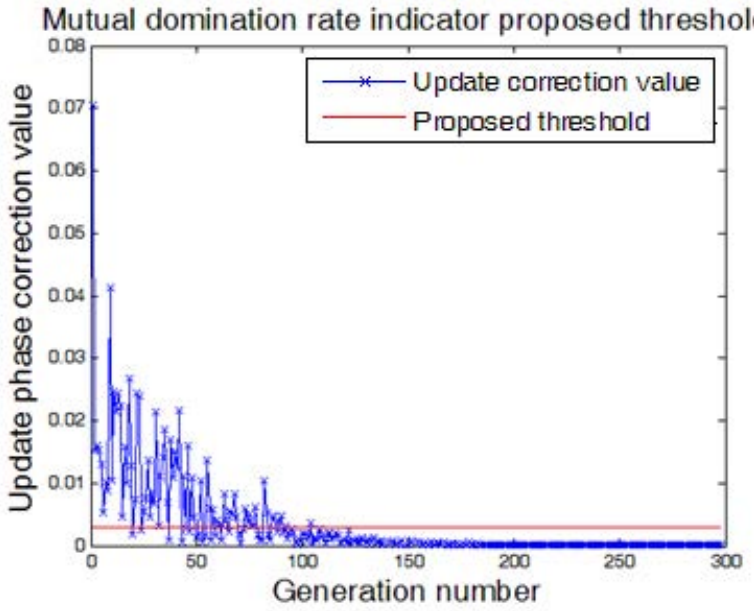
It is important to remember that, even though the distance evolutions to the optimal front may look similar for different executions, the shown data are an estimation of that distance (in our case using the hypervolume indicator) and we are also using an estimation (in fact, three different estimations, each corresponding to one of our progress indicators) to determine when are reaching the steady final section of that distance evolution, and thus the algorithm should be stopped.

Similar figures of the distance evolution to the optimal Pareto front may have different progress indicator's figure, and thus trigger our stopping criterion at different generations. This reassures the importance of using some data fusion architecture over our individual indicators, to be able to get results less problem and execution dependent. In any case, the stopping generations obtained are very close to the expected optimal stopping generation, allowing an overall satisfactory consideration over the obtained results and the presented criterion execution independent.

Analysis with different test functions and MOEA's

In this section the test cases will be extended to the different problems and algorithms sets previously defined, using the same parameters for the filter and thresholds already presented, in order to test its performance in different environments. Table 5.4 shows the results for the chosen set of algorithms and problems. Also, figures 5.17 and 5.18 present sample evolutions for the alternative problems (DTLZ6 and DTLZ7) and algorithms (SPEA2 and PESA) not

Figure 5.13: MDR indicator sample evolution along with proposed threshold



covered in previous figures.

Table 5.4 shows the results for only one execution of the stopping criterion with the algorithms and problems indicated. A wide range of experiments was performed, but it was considered best to include only a single execution instead of the mean stopping iteration, as they were independent, for comparative purposes among the different alternatives. The chosen executions are relevant in terms of showing the effects that could be appreciated in the whole experiment set. The different performance over the different algorithms can be caused by the inexactitude in choosing the R and Q parameter values (as previously explained, this is a MOP itself). Also different situations arise and the different progress indicators do not share the same sensibility characteristics to be able to detect the proper stopping generation.

As presented in figure 5.10, Kalman filtering is applied to obtain the individual stopping generation for each of the indicators, and the final stopping generation of our algorithm is reached when two of those indicators have reached theirs (a simple decision fusion architecture). This architecture also resembles the least squares data fusion architecture presented in section 5.2.3, but instead of whole navigation solutions, the output of the isolated sensor processors is the decision of whether the algorithm should be stopped.

The proposal of this experiments is not to compare the individual performance of the indicators, but rather to test the validity of the proposed stopping criterion. The validation difficulties already exposed for previous cases arise. Figures 5.14-5.18 attempt to achieve a graphic summary of the results. One important aspect not included in this comparison is the running computational time needed for each of the generations, as it is increased by the processing of each of the individual indicators (some of them, as the hypervolume, with a high computational cost). This chapter has been focused on the final stopping iteration and its accuracy.

Figure 5.14: Stopping generation comparison for DTLZ3, NSGA-II, first execution

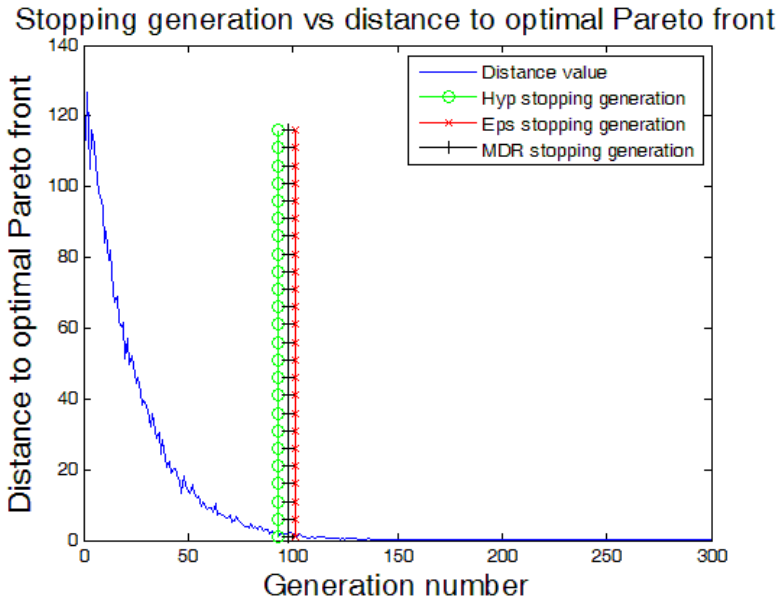


Figure 5.15: Stopping generation comparison for DTLZ3, NSGA-II, second execution

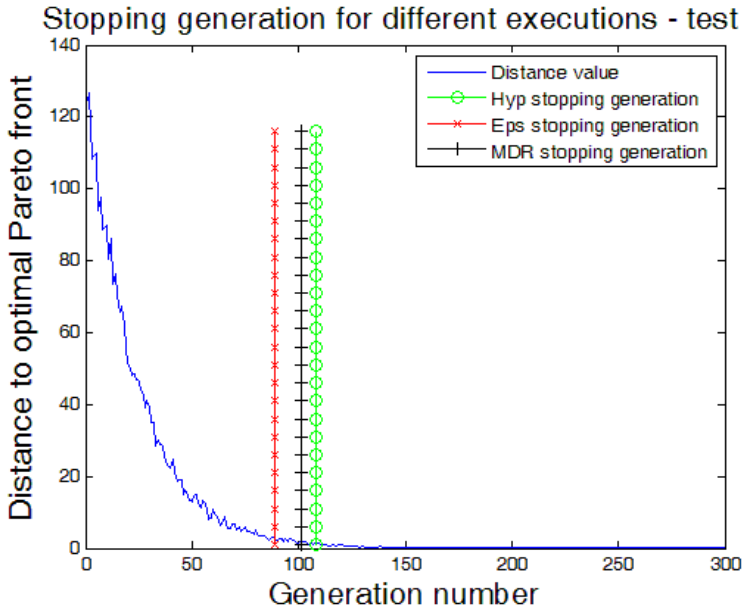


Figure 5.16: Stopping generation comparison for DTLZ3, NSGA-II, third execution

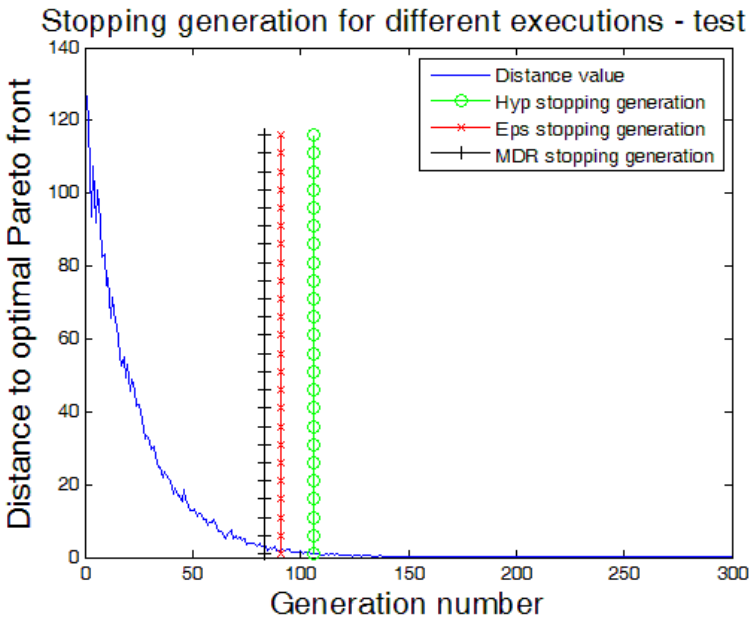


Table 5.4: Test results for the proposed problem and algorithm sets

Algorithm /Problem	Indicator / Stopping generation			Stopping criterion generation
	Hypervolume	Epsilon	MDR	
NSGA-II / DTLZ3	93	101	98	98
SPEA2 / DTLZ3	84	82	63	82
PESA / DTLZ3	104	99	88	99
NSGA-II / DTLZ6	110	85	95	95
SPEA2 / DTLZ6	98	80	80	80
PESA / DTLZ6	105	98	95	98
NSGA-II / DTLZ7	168	137	171	168
SPEA2 / DTLZ7	154	142	147	147
PESA / DTLZ7	206	177	115	177

Figure 5.17: Stopping generation comparison for DTLZ6, PESA

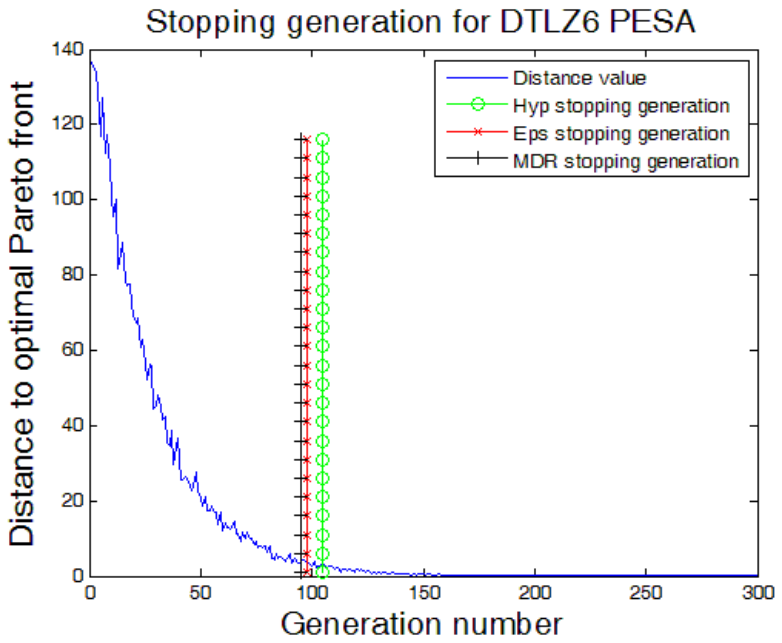
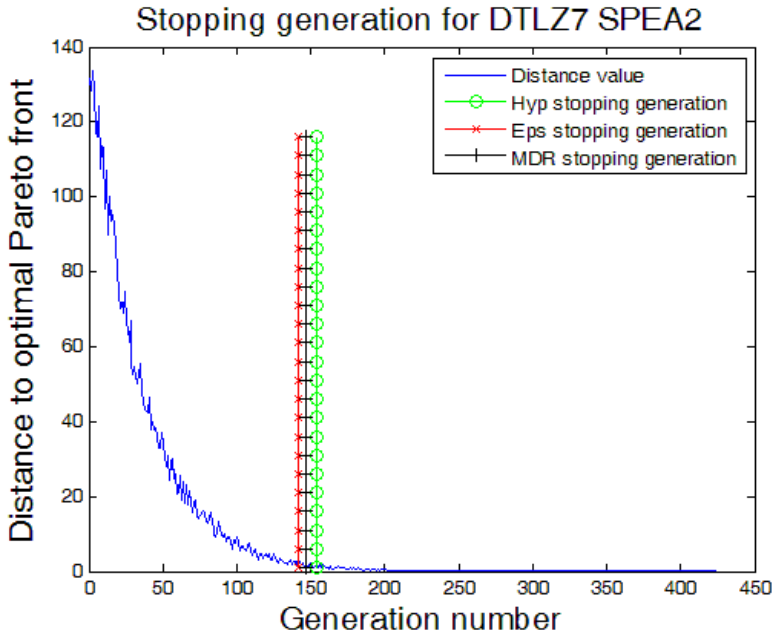


Figure 5.18: Stopping generation comparison for DTLZ7, SPEA2



5.3 Focusing on simplicity and efficiency: the LSSC Criterion

Section 5.2 has introduced a stopping criterion based on linear estimation of several quality indicators, according to a Kalman filter, which are taken into a decision fusion system to determine whether the evolutionary algorithm must be stopped. The experimental validation in 5.2.4 have shown interesting results, regarding the applicability of the different quality indicators used and the overall criterion. Also, a fact has been pointed out: there is a lack of optimal pre-establish stopping generation (or number of function evaluations) to measure the quality of a stopping criterion. In any case, results showed that, based on a pre-configured desired stopping threshold, the algorithm could be configured for a specific problem instance and algorithm and those configuration parameters remained robust for different algorthms and problems.

The objective of this section is to provide a stopping criterion which simplifies the configuration using a more restricted linear estimation while, at the same time, focuses on the efficiency of the proposal. This proposal will basically eliminate the model and measuring noise configurations, proposing an analysis of a specific window of the values of the quality indicators previously introduced. Even though it will not be detailed again, the fusion architecture proposal of section 5.2.3 could be reapplied, since LSSC will propose and alternative method only for the individual indicator's stopping criterion.

5.3.1 Global stopping criteria

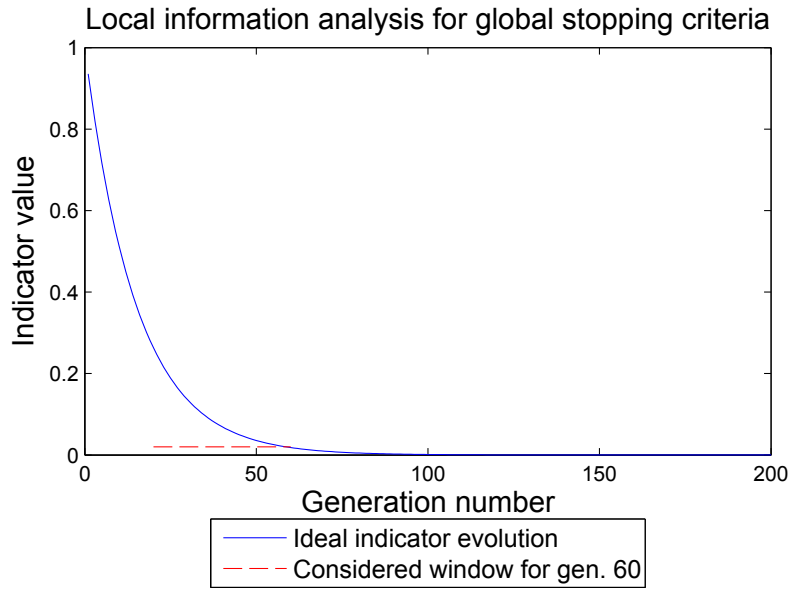
As seen in the domain techniques analysis in section 2.7 and also in the proposed technique in section 5.2, the most common approach for stopping criteria in multi-objective evolutionary algorithms is to identify when the evolution of the indicator becomes linear, where the tendency of that evolution (or, in other words, the amount of improvement over the solution per generation) falls below a certain threshold (or is considered irrelevant). This threshold is used as an application of the stopping scenarios presented, and the linear check determines the validity of the obtained tendency.

Obviously, the complete evolution of the indicator never follows that criterion (if it did, we would never reach an acceptable solution) so the name *global criteria* may be a misleading one, due to the fact that the algorithm will only be looking at a local portion of the indicator evolution each generation (or considering it as Markov process, such as in Kalman approaches). This analysis window usually covers the value of the indicator for a certain amount of previous generations (which may change its value dynamically). Figure 5.19 shows this process.

In the example in Figure 5.19, the evolution of the indicator is clearly non-uniform in the considered window, and thus the criterion would probably determine that the MOEA must continue running. That idea can also lead us to some considerations about the window size these algorithms must have in order to prevent inaccurate stops. An example of such an stopping situation is provided in figure 5.20.

The example in Figure 5.20 considers a window of the same size as Figure 5.19 (40 generations). With this size, regardless of the concrete technique applied, probably any generation from 65 to 75 would be considered to stop, even though, seeing the whole indicator evolution, we may notice that there are improvement chances after that stagnation of the indicator value. A bigger window size (depending on the technique applied, probably for 55 or

Figure 5.19: Example of local information analysis performed as part of a global stopping criterion



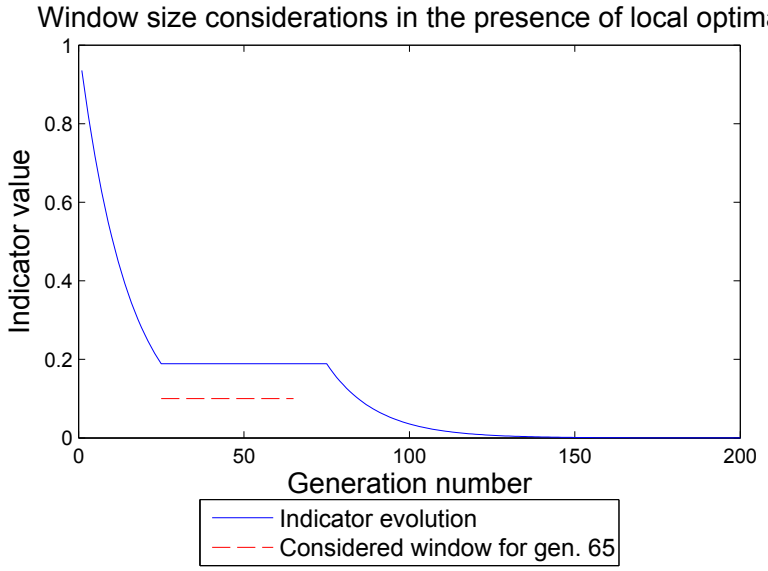
more generations) would have been able to determine that the indicator, even though it had become very stable, was improving again. This situation shows that, if the problem is known or suspected to have local optima or different situations (such as neutral drift scenarios) where the indicator may reach a stagnation situation, the considered window size must be considerably larger, in order to be able to determine correctly if the improvement over the indicator value has stopped permanently.

General stopping situations for evolutionary algorithms were covered in section 2.4.7. Specifically for the multi-objective case, these situations have been reviewed in (Martí et al., 2009), according to the following criteria

1. The amount of computation is sufficient.
2. A solution obtained so far is satisfactory.
3. The solution is not satisfactory, but a better one is unlikely to be produced.
4. The method is not able to converge to a solution.
5. Additional computation will provide little or no improvements in the current solution.

The first situation was the one covered by traditional approaches, where the amount of computation was measured in number of generations (exhaustion-based criteria). There have been modifications based on the number of function evaluations and the introduction of modeling methods, as an approach to more complete convergence criteria, but they still needed a number of fixed function evaluations, which could be high (30000-500000) when the key parameter was the quality in the approximation (Deb et al., 2003) or low (130-250), in the case of model-assisted approaches (Knowles, 2006). One of the handicaps of this

Figure 5.20: Example of unsuccessful stopping analysis at local minima



approach is that it may be dependent on several different parameters, such as the population size, the selection technique, the complexity of the fitness functions, etc.

The second situation is the one approached by quality indicators, which requires, as was pointed out in the introduction, to know the OPF a priori (in order to be applied automatically) or a decision maker which can estimate the quality of a given solution (reference criteria). The automatic application of this criterion involves the knowledge of the solution prior to the application of the algorithm, making it inapplicable to resolve new problems for which the OPF is not known (relegating this criterion to validation and comparison issues rather than using it as a general tool). On the other hand, a decision maker usually also requires a good knowledge over the expected solution in order to test the validity of a given Pareto front.

The differences between the following three situations are very subtle. The third situation presents a scenario where the solution value has converged but is not satisfactory, in the fourth one the solution is no able to converge, while the fifth implies that the evolution in the improvement over different solutions at different generations has become not significant for the algorithm. Quality indicators are required in order to automatically determine if the solution is unsatisfactory, but at the same time, some analysis of the progress towards the solution is required to determine if a better one is likely to be obtained or not.

With these requirements, knowing the optimal Pareto Front would be required as well, but looking at the fifth situation, the reader may consider whether it is needed to know if the current solution is satisfactory or not if no further improvement can be gained by the application of the algorithm. This point of view allows us to summarize the three final scenarios into obtaining the best solution possible without considerations about its quality, at least as part of the stopping criteria (considering the focus of quality indicators, these three cases could be considered as distribution and movement based stopping situations). However, third and fourth situations introduce novel considerations, since the fact of unsuccessful final solutions is more linked to multi-objective optimization, particularly to those cases which

deal with many objectives. The MGBM criterion (section 2.7.2 is particularly focused on the detection of these stopping situations.

An interesting parameter for general convergence criteria is the analysis of the computational complexity added by the stopping criterion used. The complexity of the indicators used is usually high enough to clearly exceed (and thus eliminate in the typical complexity order analysis) the one added by the stopping criterion. This is especially true when dealing with quality indicators such as hypervolume. Also, the complexity of the overall evolutionary algorithm is usually dominated by the cost of the function evaluations, which has lead to the use of indicator based evolutionary algorithms (such as IBEA (Zitzler & Künzli, 2004)).

5.3.2 Data gathering and processing: the least squares stopping criterion

The objective of the Least Squares Stopping Criterion (LSSC) proposed in this section is to introduce a stopping criterion which can be easily implemented in any programming language (to facilitate the task of incorporating it to any MOEA) and easily configured by parameters related to MOEA's research (instead of those related to the particular techniques applied by the criterion).

The underlying idea is to determine when our progress indicator has reached a stagnation situation. To achieve this, we will base our criterion on two different considerations: the adjustment to a uniform model (by means of a least squares approximation, which gives its name to our criterion) and the value of the slope of that uniform model (as a measure of the progress between two different generations).

Simple Least Squares (Meyer, 1970) is a basic linear regression method which approximates a variable according to the model presented in equation 5.20.

$$y = a + bx \tag{5.20}$$

It introduces some key assumptions, among which some of the most important are zero mean error and constant variance in the indicator value. These assumptions, which make this approach inapplicable to some real problems, fit our stopping needs (these are the circumstances under which we would like our algorithm to stop). It is also noticeable, compared to the Kalman approach presented in section 5.2, that there is no underlying considerations for the errors introduced either by the linear model or the measuring process. Representing y the indicator value and x the generation number, a and b can be calculated with the following matrix-based formula presented in equation 5.21

$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & w_l \end{bmatrix}^{-1} * \begin{bmatrix} \sum x_i * y_i \\ \sum y_i \end{bmatrix} \tag{5.21}$$

where w_l is the length of the chosen window. Once we have determined the values of the parameters for our linear regression, we need to define the normalized residue value, presented in equation 5.22.

$$res = \frac{\sum_i (y_i - (a + bx_i))^2}{w_l} \approx \frac{\chi^2}{w_l} \tag{5.22}$$

As shown in equation 5.22, the obtained residue follows a chi-square distribution which has $w_l - n$ degrees of freedom, where $n = 2$ in our case (as we are imposing two linear

restrictions, a and b parameters). This distribution has known mean and variance, presented normalized in equations 5.23 and 5.24.

$$\mu = 1 - \frac{2}{w_l} \tag{5.23}$$

$$\sigma^2 = \frac{2}{w_l} - \frac{4}{w_l^2} \tag{5.24}$$

With the values in equations 5.23 and 5.24, we may use Tchebycheff's inequality (Meyer, 1970) to determine a threshold to leave a certain percentage of the residues coming from a uniform distribution bellow its value. The probabilistic relationship of this percentage and the mean and variance of the distribution are presented in equation 5.25

$$Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{5.25}$$

For the k parameter, the classically used value of three has been chosen. This will leave roughly the 90% of the measures belonging to a uniform distribution bellow the computed threshold. This threshold can be obtained adding three times the standard deviation to the mean value, which is presented in equation 5.26

$$thres = \mu + 3\sigma = 1 - \frac{2}{w_l} + 3 * \sqrt{\frac{2}{w_l} - \frac{4}{w_l^2}} \tag{5.26}$$

When the value of the residue shown in equation 5.22 falls below the threshold exposed in equation 5.26 we can consider that the evolution of the indicator has started to be uniform. This is a needed restriction to stop the algorithm's evolution, but may not be sufficient. An example of a stopping situation based only on this residue's value is shown in figures 5.21 and 5.22.

The complimentary restriction is the slope value, which will allow us to stop our evolution not only when it has become uniform, but also when the amount of that evolution per generation has become insufficient. Thus, the complete stopping criterion is defined with equation 5.27.

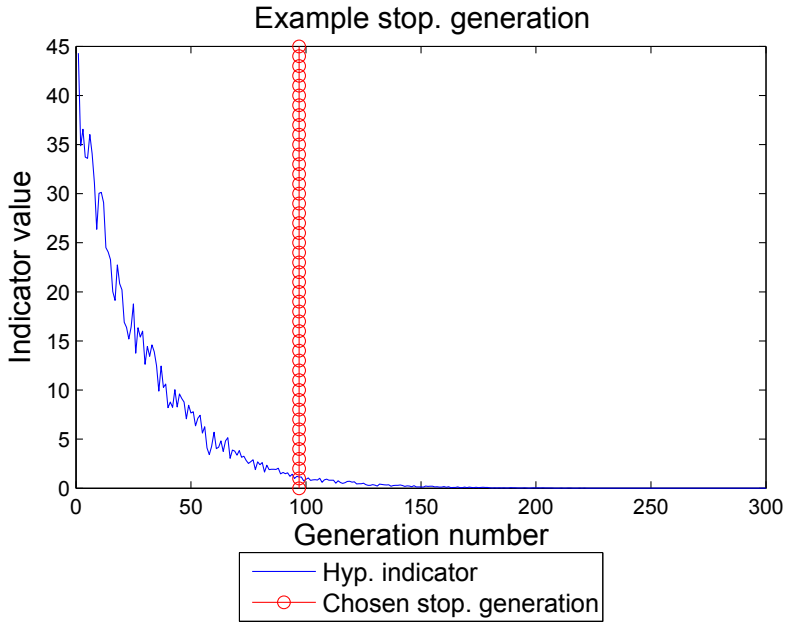
$$\frac{\sum_i (y_i - (a + bx_i))^2}{w_l} < 1 - \frac{2}{w_l} + 3 * \sqrt{\frac{2}{w_l} - \frac{4}{w_l^2}} \bigwedge b < min_{prog} \tag{5.27}$$

where a and b are computed with equation 5.21.

The process to choose the min_{prog} value is rather simple: the researcher chooses a number of examples, determines which would be the ideal stopping generation for them and tunes the value of the min_{prog} parameter to guide the stopping criterion to stop at the chosen generation. This parameter value can then be applied to new problems and/or algorithms without requiring any modification in it (as it will be shown in the experimental section), with a similar behavior to the parameters from the Kalman approach presented in section 5.2.

For the given example in figures 5.21 and 5.22, which shows hypervolume indicator applied over a NSGA-II MOEA to the DTLZ3 problem, we have chosen an absolute value for the min_{prog} parameter of 0.002 (in fact this will be the used value for the parameter in the experimental section whenever we are using the hypervolume indicator, regardless of the

Figure 5.21: Overview of a stopping criterion instance based only on residue's value



problem or the algorithm). Figures 5.23 and 5.24 show the behavior of the stopping criterion, as presented in equation 5.27, over the same example.

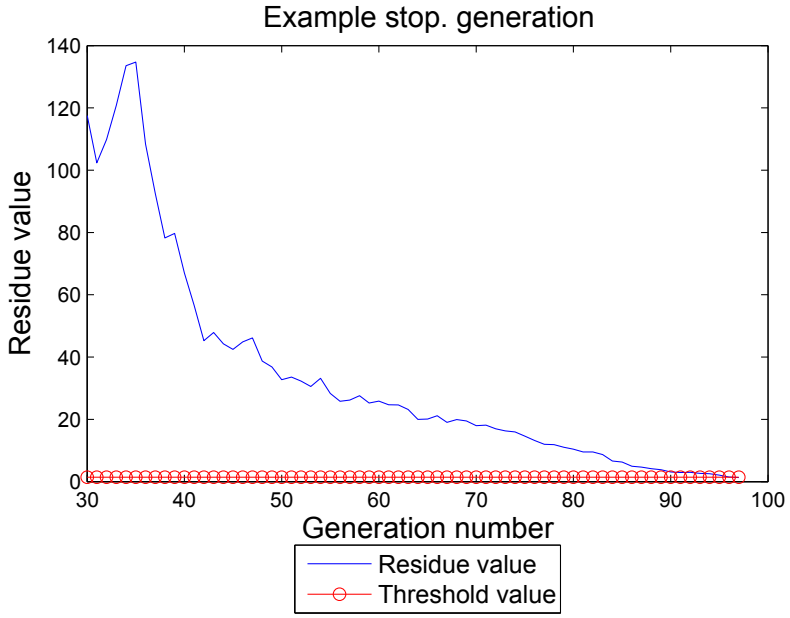
The chosen value for the slope is fairly conservative (in fact, more than the configuration values presented for the Kalman approach in section 5.2.4), trying to stop only when future improvements would be marginal ones. Some users may prefer higher values, sacrificing small indicator improvements in favor of faster results. The dependency with the window size parameter, in absence of problematic local situations such as the ones presented in the global stopping criteria section, is not strong, only increasing, according to its value, the final stopping generation once the evolution of the indicator's value starts to satisfy the criterion conditions.

It is important to remember that this stopping criterion is not necessarily applied on its own for a given algorithm: very usually, along with it, the user will add a different one regarding the maximum computation allowable for the algorithm (either in time, number of generations or number of function evaluations), constituting a combined criterion, as introduced in section 2.4.7. Also, LSSC is a substitute only for the linear estimation of the Kalman approach previously presented, which implies that the voting fusion architecture introduced in section 5.2.3 can be used along with it.

5.3.3 Complexity analysis

LSSC uses a window-based procedure to gather the quality indicators values, similarly to the OCD algorithm (presented in section 2.7.1). However, the way in which this data is collected differs in the way in which the binary quality indicators are applied, heavily impacting the computational complexity. Figure 5.25 shows the application used by OCD to recompute its indicators' values every time a new generation is generated (and, thus, a new Pareto front is formed).

Figure 5.22: Overview of residue value analysis during an example evolution instance



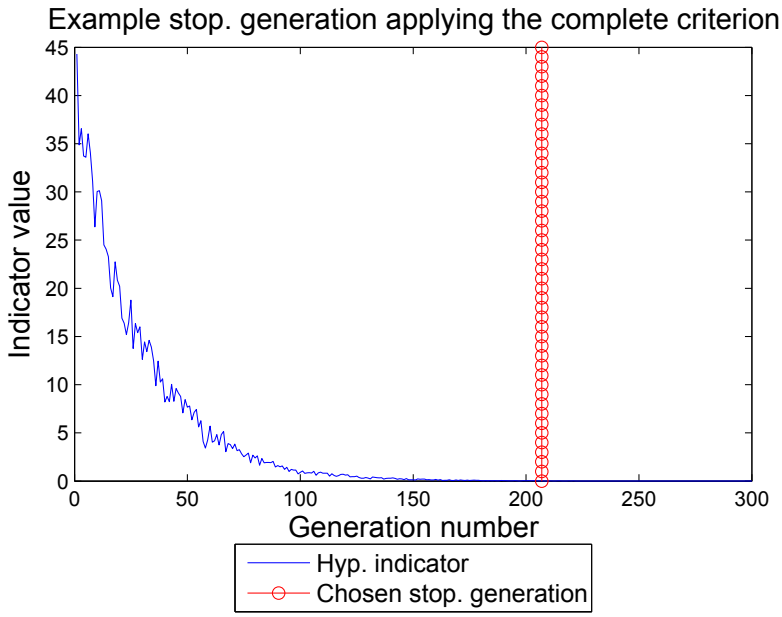
As seen in figure 5.25, every time a new generation is processed in order to test the stopping criterion, a whole recomputing of the quality indicators takes place, comparing each of the previous generations fronts to the one newly introduced in the window. Even for the unary hypervolume indicator used, this recomputation takes place, since a new normalization is introduced at each step. This increases the computational cost of the algorithm, and also makes it linearly dependent with the window size.

LSSC, however, computes its binary quality indicators between consecutive generations, requiring only one additional set of indicators every time a new generation is produced. This process is shown in figure 5.26.

The fact that the window size, w_l does not affect the computational complexity of the algorithm may allow higher values of this parameter to be used in the configuration, in order to prevent possible situations such as the one shown in figure 5.20. Apart from the data gathering process itself, it is interesting to analyze the complexity of the processing of this data. At every generation (excluding the first $w_l - 1$, which form the first generational window) the algorithm has to obtain the following values: the linear regression parameters (a and b), the threshold value ($thres$) and the residue value (res). It is useful to invert the order in which the conditions are checked in equation 5.24, in order to obtain the res value only if necessary. For a fixed window size, the threshold value only has to be computed once, applying the same value for all the comparisons. Using standard libraries, both the regression parameters and the residue value can be obtained in $O(w_l)$, being this the complexity order of the criterion.

In equation 5.21 we intentionally presented the required formula for the calculation of the two required linear regression parameters, due to the fact that, being composed of different summations which only differ in one term for consecutive generations, once the initial parameters have been calculated, the rest can be obtained with a constant order complexity. Equation 5.28 shows this iterative computation of the least squares approximation parameters

Figure 5.23: Overview of the proposed stopping criterion



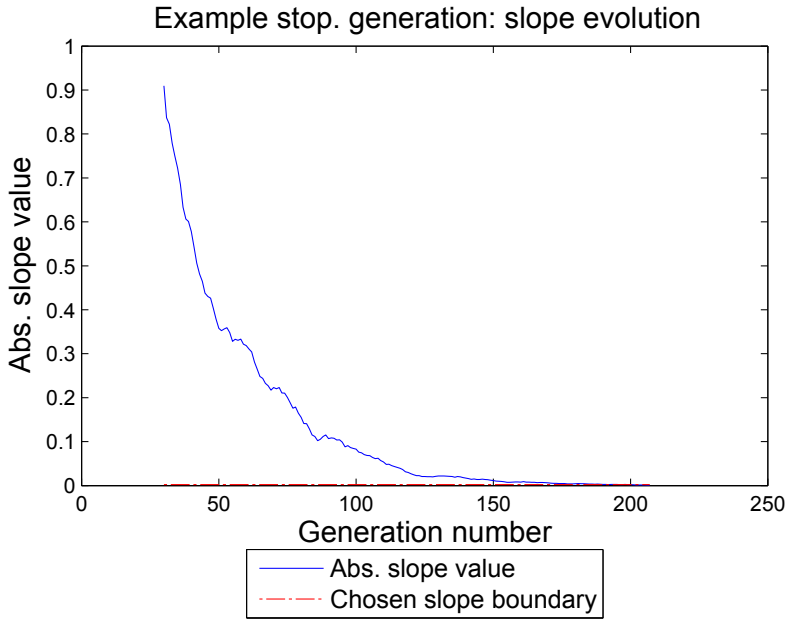
$$\begin{bmatrix} b_i|b_{i-1} \\ a_i|a_{i-1} \end{bmatrix} = \begin{bmatrix} (\sum x_{i-1}^2) - x_{i-w_l}^2 + x_i^2 & (\sum x_{i-1}) - x_{i-w_l} + x_i \\ (\sum x_{i-1}) - x_{i-w_l} + x_i & w_l \end{bmatrix}^{-1} * \begin{bmatrix} (\sum x_{i-1} * y_{i-1}) - x_{i-w_l} * y_{i-w_l} + x_i * y_i \\ (\sum y_{i-1}) - y_{i-w_l} + y_i \end{bmatrix} \tag{5.28}$$

This does not change the worst case complexity order of the stopping criterion (the *res* value still has to be calculated in $O(w_l)$) but, by checking the min_{prog} condition first, it allows the criterion to run in $O(1)$ most generations, without requiring a complex implementation, and becomes specially interesting and advisable when the progress indicator is also required by the selection criterion (and thus integrated in the MOEA's usual cycle) or computationally inexpensive to calculate (such as MDR).

5.3.4 Experimental validation

The dataset configuration is going to be the same presented in section 5.2.4: three different algorithms (NSGAII, SPEA2 and PESA) and three different problems (DTLZ3, DTLZ6 and DTLZ7). For the optimum configuration of the stopping generation factors such as the cost of running additional generations or the required accuracy in the final solution may be considered, but only the progress indicator used is required to be analyzed in order to determine the right slope value. Results regarding the three different presented progress indicators will be shown in this section, so three different slope values need to be configured. The chosen values are 0.002 (hypervolume) 0.0004 (epsilon) and 0.00002 (MDR).

Figure 5.24: Overview of the slope value analysis applying the full stopping criterion



These values have been chosen according to the process explained in the method's presentation section. Intuitively, these values can be related to the different ranges which the indicator exhibit. The window size will be constant regardless of the indicator used, and fixed at 30 generations (this value is the minimal number of measures to assume normality in the distribution). To determine the quality of the stopping generation obtained, we will compute the hypervolume difference compared to an a priori fixed generation (which will be chosen based on the problem's difficulty, according to the values suggested in the datasets or algorithms reference papers).

Each experiment has been run thirty times. We provide the statistical values of mean, standard deviation, minimum and maximum obtained (both for the stopping generation and the hypervolume difference with the a priori stopping generation), to verify the criterion performance. Tables 5.5-5.7 show these results. Also, figures 5.27-5.29 show the results of the stopping generation obtained for the different algorithms and problems. Finally, the comparison of the obtained stopping generation compared to the binary hypervolume value of the stopping generation versus the final generation considered is presented in figures 5.30-5.32.

The chosen values for the slopes were quite conservative, in order to obtain very accurate results. This can be observed in that the maximum hypervolume difference in mean value is 0.14, whereas in maximum value is 0.331 (both obtained for the hardest problem, DTLZ7). Even so, we have managed to obtain a stopping generation whose value is about 2/3 of its respective a priori one, with significantly similar performance results over the different indicators.

Table 5.5: Stopping criterion results for the DTLZ3 problem

Alg	Ind	Stopping generation					Hypervolume difference			
		Mean	Min	Max	Std D	A priori	Mean	Min	Max	Std D
NSGAII	eps	204,733	193	215	5,9186	300	0,027	0,011	0,057	0,012
NSGAII	hyp	196	184	207	5 79536	300	0 04	0 016	0 073	0 015
NSGAII	MDR	219 233	202	233	7 93371	300	0 015	0 004	0 038	0 008
PESA	eps	203 733	188	213	6 10219	300	0 049	0 016	0 108	0 021
PESA	hyp	210 533	194	223	7 394	300	0 046	0 015	0 096	0 02
PESA	MDR	200 367	185	212	6 58359	300	0 055	0 027	0 11	0 022
SPEA2	eps	178 267	165	188	4 63073	300	0 024	0 007	0 056	0 013
SPEA2	hyp	174 033	166	182	4 27892	300	0 032	0 015	0 059	0 012
SPEA2	MDR	174 567	167	184	3 92765	300	0 027	0 011	0 06	0 014

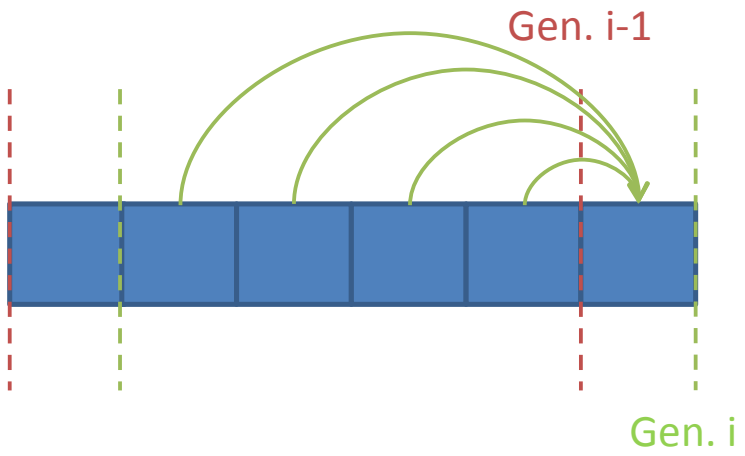
Table 5.6: Stopping criterion results for the DTLZ6 problem

Alg	Ind	Stopping generation					Hypervolume difference			
		Mean	Min	Max	Std D	A priori	Mean	Min	Max	Std D
NSGAII	eps	208 233	192	220	6 1346	300	0 026	0 01	0 054	0 012
NSGAII	hyp	199 567	183	211	6 22389	300	0 045	0 017	0 081	0 015
NSGAII	MDR	225 033	205	239	9 7432	300	0 017	0 004	0 048	0 011
PESA	eps	207 333	194	222	6 61937	300	0 053	0 02	0 101	0 023
PESA	hyp	213 467	203	231	5 74596	300	0 044	0 018	0 087	0 017
PESA	MDR	205 633	198	215	4 25468	300	0 057	0 024	0 094	0 02
SPEA2	eps	183 767	176	193	4 11627	300	0 02	0 006	0 044	0 009
SPEA2	hyp	179 367	174	186	3 36804	300	0 032	0 014	0 053	0 009
SPEA2	MDR	177 1	166	185	4 24548	300	0 03	0 014	0 049	0 01

Table 5.7: Stopping criterion results for the DTLZ7 problem

Alg	Ind	Stopping generation					Hypervolume difference			
		Mean	Min	Max	Std D	A priori	Mean	Min	Max	Std D
NSGAII	eps	302 733	279	322	12 4123	425	0 094	0 033	0 181	0 041
NSGAII	hyp	296 8	263	316	11 583	425	0 113	0 049	0 215	0 041
NSGAII	MDR	332 233	300	368	16 3532	425	0 047	0 004	0 109	0 027
PESA	eps	301 067	264	322	12 956	425	0 14	0 052	0 331	0 065
PESA	hyp	313 867	288	337	12 125	425	0 134	0 073	0 244	0 039
PESA	MDR	310 433	276	335	12 7135	425	0 113	0 045	0 257	0 051
SPEA2	eps	262 6	243	275	8 47145	425	0 082	0 039	0 153	0 034
SPEA2	hyp	265 567	245	294	10 8522	425	0 093	0 025	0 17	0 035
SPEA2	MDR	268 5	246	279	8 88916	425	0 069	0 03	0 107	0 023

Figure 5.25: Quality indicator values gathering in the OCD algorithm for every generational window



5.4 Conclusions

The intention of this chapter was to accomplish the proposal of a stopping criterion using several indicators available in the community with an estimation theory based schema, allowing the accumulation of evidence from these indicators until a certain pre-established threshold is trespassed and the stopping criterion triggered. Two different proposals have been presented regarding the individual processing of indicators data, one based on Kalman filtering and the other based on a simpler least squares approximation.

Kalman filtering proposal has introduced an alternative to an existing approach in the literature which had simplified the filtering process, and also highlighted the issues related to the model and measuring processes noise. Instead of tracking the filtered output, the approach is based on the tracking of the filter corrections, trying to determine the linearity of the result according to the model used. Along with these filtering estimation, several data fusion architectures (extracted form sensor fusion approaches) have been reviewed to cope with the stopping problem in MOEA's and the input data from the chosen indicators. Issues regarding the lack of proper formalization and theory regarding the different quality indicators has been remarked, leading to a simple data fusion (decision fusion), which was carried through the experiment phase, showing in spite of its simplicity the power of these architectures to help us avoid problematic concrete situations for individual indicators.

Experimental results show that a concrete stopping criterion configuration, once established, can be ported to different problems and algorithms. On the other hand, one of the main difficulties regarding the introduction of a MOEA stopping criterion is presented: the lack of an optimal stopping generation (even for pre-established algorithms and/or problems). This makes it extremely difficult to analyze the quality of a set of results regarding its stopping generation, or the comparison between different algorithms for such an issue, since additional generations, for almost every practical application case, involve an increase (even if small) in the solution quality (in the absence of genetic drift). The results presentation was focused, according to this difficulty, in the robustness of the proposed approach, the results

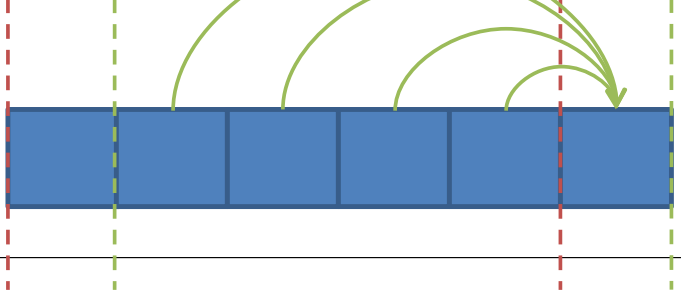


Figure 5.26: Quality indicator values gathering in the LSSC algorithm for every generational window

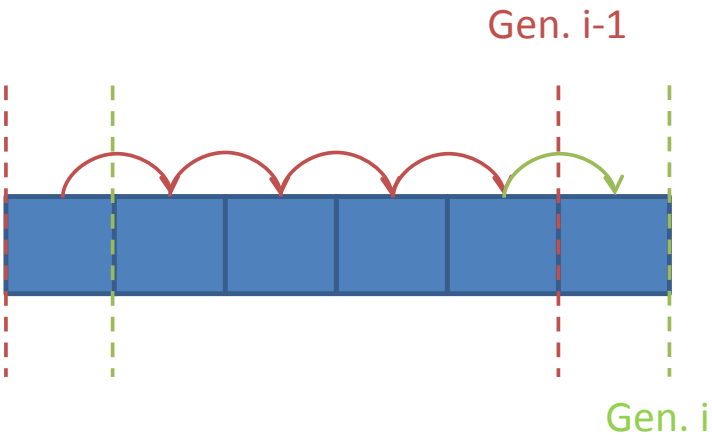
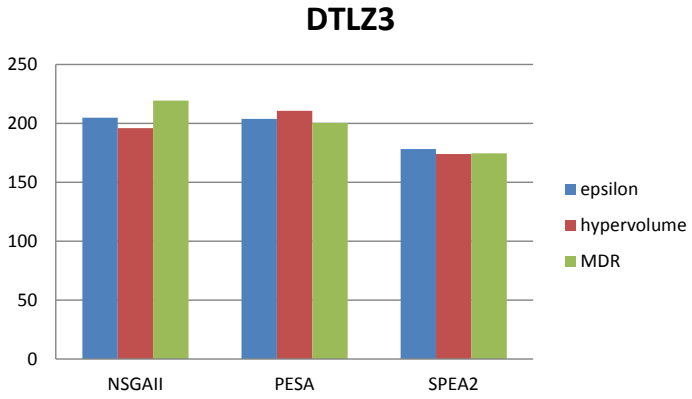


Figure 5.27: Stopping generation results for DTLZ3 problem



regarding the voting system, and a graphical visualization of the results to compare them to the objectives proposed, which were satisfactory.

LSSC, Least Squares Stopping Criterion, the final proposal of this chapter, has its basis on the previous results obtained with the Kalman approach along with some observations gathered from the literature regarding stopping criteria. The main consideration was the lack of an established stopping criterion for MOEAs, opposed to the single objective case (where a simple tracking of the best individual is a commonly applied approach and usually included in the associated frameworks). The two main reasons for this fact were considered: complexity of available approaches (which imply complex filtering systems or heavy statistical testing) and the lack of statistically sound comparison approaches to determine the quality of the different approaches. In order to face these issues, LSSC attempts to provide an easily configurable and implementable criterion, being at the same time robust and efficient.

LSSC is an alternative to the estimation performed by the Kalman filter approach previously presented (and can be used along with the proposed voting fusion system). It establishes

Figure 5.28: Stopping generation results for DTLZ6 problem

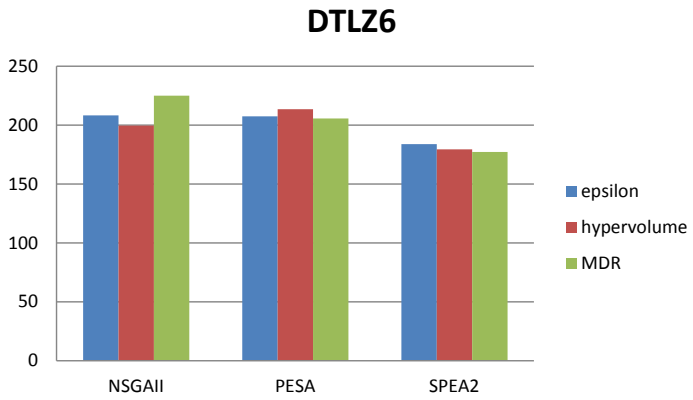
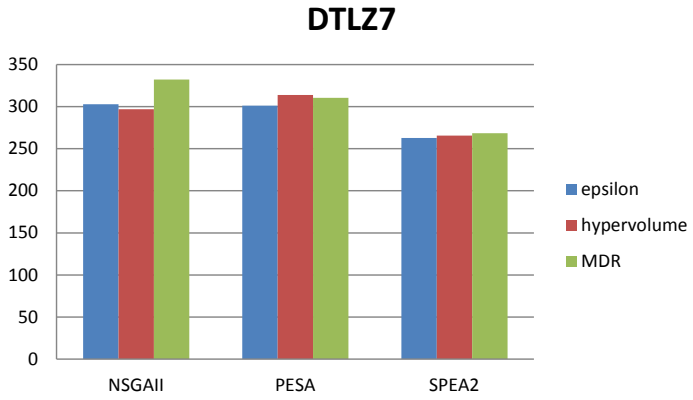


Figure 5.29: Stopping generation results for DTLZ7 problem



its stopping decision upon two different facts: the normality of the results analyzing binary quality indicators between consecutive generations and the slope of the resultant least squares approximation. The stopping decision can be computed using a simple logical equation, based on two parameters: the number of generations analyzed and a threshold which depends on the concrete quality indicator used. The value for the number of analyzed values is fixed on 30, and the threshold can be controlled by the researcher in order to suit its quality needs. Even so, threshold values are proposed for each of the three quality indicators used. These suggested configuration values can be used to provide a black-box stopping criterion.

Also, the presented data gathering only requires one update over the quality indicators values, being this update that of applying the binary indicator from the previous generation Pareto front to the one newly included into the generational window. To contribute to the computational efficiency, an iterative computation of the least squares approximation parameters has been presented, allowing the stopping criterion to run in a constant complexity order for most generations (those that do not meet the maximal slope condition).

Experimental results for LSSC, while still suffering from the lack of a known optimal

Figure 5.30: Stopping generation vs hypervolume for DTLZ3 problem

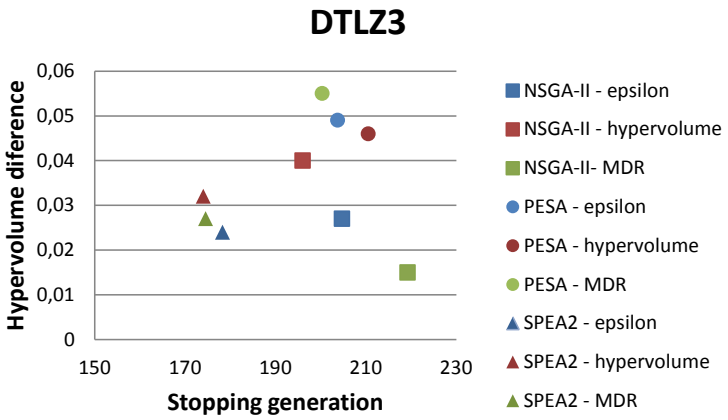
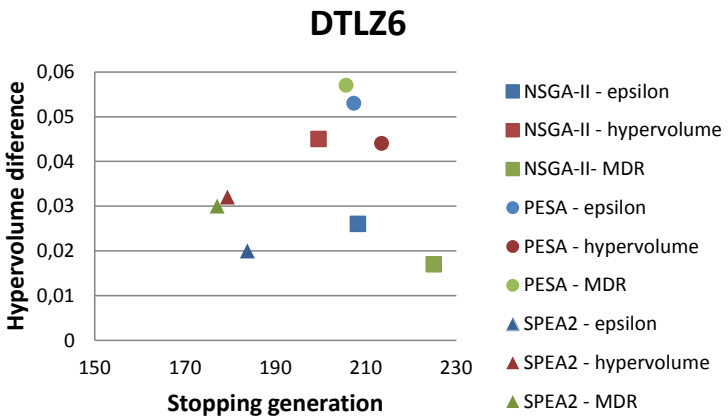


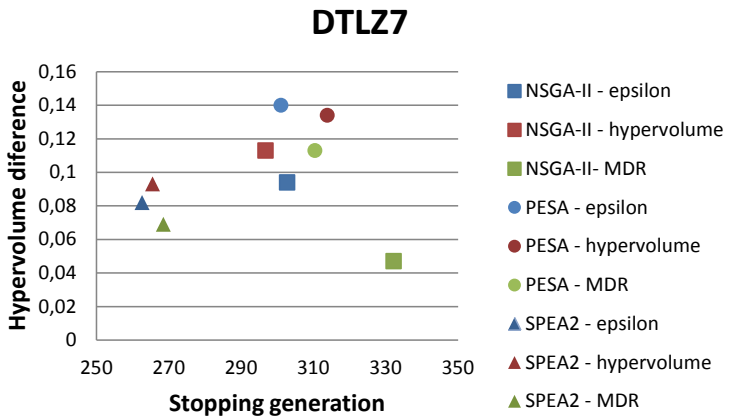
Figure 5.31: Stopping generation vs hypervolume for DTLZ6 problem



stopping generation, have been focused on the comparisons between the stopping generation and some pre-established future stopping generation in terms of hypervolume, to assess the quality of the stop, as an alternative to the more robustness oriented data presentation of the Kalman configuration, which also relied on graphical validation. These results show the robustness of the configuration parameters through different algorithms and test problems, similar to the obtained with the more complex Kalman approach.

Following the thesis objectives, this chapter has analyzed the difficulties of stopping criteria for multi-objective evolutionary algorithms, as opposed to chapter 4, where the single-objective case we faced. The proposed criterion is based on the values of progress indicators (binary indicators which measure the improvements between two different generations) and a least squares regression to detect the stagnation situation, providing a simple and effective solution which can be easily integrated as a black box into any available algorithm.

Figure 5.32: Stopping generation vs hypervolume for DTLZ7 problem



6

Multiobjective evolutionary polygonal approximation

> *There exists a world In terms of probability this borders on the impossible*
> *It would have been far more likely if, by chance, there was nothing at all*
> *Then, at least, no one would have began asking why there was nothing*

Jostein Gaarder, *Maya*, 1999

This chapter presents and formalizes an explicit multi-objective evolutionary approach for the segmentation issue according to Piecewise Linear Representation, which consists in the approximation of a given digital curve by a set of linear models minimizing the representation error and the number of such models required. Available techniques are focused on the minimization of the quality of the obtained approximation, being the cost of that approximation considered, in general, only for certain comparison purposes. The multi-objective nature of the problem is reviewed (it was initially considered in chapter 3, where it was required to include quality indicators as the comparison method for the final results) and its treatment in available works analyzed, presenting an a-posteriori approach based on an evolutionary algorithm. Three representative curves are included in the data set, comparing the proposed technique to nine different techniques. The main references for this chapter are (Guerrero et al., 2012, 2010a, 2012b).

6.1 Introduction

Digital curves domain, leaded by the importance of human processing and understanding of visual information, established its roots with the psychological studies performed in the middle fifties (Attneave, 1954). One of the main keys to the study of this domain is the representation performed over the original data. The goal of this representation is to cover the main characteristics of a given shape with the least amount of data. This dimensionality reduction performs several objectives. On the one hand, it reduces the storage capacity required for the obtained time series, and, on the other hand, it has an immense impact on the efficiency of the subsequently applied methods, such as feature extraction (Mörchen, 2003).

Segmentation processes may resort to different representations, being Piecewise Linear Representation (PLR, also named Piecewise Linear Approximation, PLA, or polygonal approximation) among the most extended options. This scope has been deeply analyzed and used according to a data mining perspective (Gionis & Mannila, 2005; Keogh et al., 2003; Liu et al., 2008) and also as a digitization method (Marji & Siy, 2003; Sarfraz, 2008). Several works have detailed the characteristics of PLR segmentation which have led to its extensive use: simplicity, locality, generality, compactness and ease of us e (Keogh et al., 2003; Sarfraz, 2008). PLR segmentation is based on the approximation of a curve (or, more generally, a certain time series) T with length n by means of a set of K segments (where $K << n$), approximating each of these segments by a linear model. It can be also described as the process of searching the *dominant points* of a given curve, being these points the edges of the segments in the previous definition.

Polygonal approximation techniques are offline segmentation processes (since they require the whole curve they will be applied to) which can be divided into three different categories: sequential approaches, split and merge approaches and heuristic search approaches. Sequential and split and merge approaches have a strong dependency on the initial steps of their algorithms (either in the form of the starting point for the scanning or the initial segmentation performed). The outcome of these methods is extremely sensible to their segmentation criterion parameters (such as error tolerance), values which may not be easy to determine. On the other hand, heuristic based approaches are computationally expensive, being not guaranteed to be optimal.

Most of the different presented techniques share the lack of a direct mechanism to control the number of segments obtained (and through it, the compression performed over the original data), even though indirect mechanisms may exist (e.g., error tolerance indirectly controls segment length, which along with the number of elements in the original data determines the number of segments in the final representation). Other alternatives, such as evolutionary approaches, allow the choice of the number of segments but lose the control over the approximation error. Comparisons between different algorithms, especially in the data mining domain (Keogh et al., 2003) are usually performed according to the error value obtained by the representation, not considering the cost of that error. Some techniques do take into account the number of segments of the obtained representation (such as in (Ray & Ray, 1992), where each cycle tries to obtain the longest possible segments with the lowest possible error value) but, since those objectives are in conflict, it is performed by what, in the multi-objective community, is usually referred to as a-priori techniques: in order to deal with different objectives in conflict jointly, a decision maker (DM) determines the importance of each of the objectives and, according to that importance, their joint value is calculated and used by underlying algorithms(Coello et al., 2007).

The previous argumentation introduces segmentation as a multi-objective optimization problem (MOOP, (Coello & Lamont, 2004)): segmenting a digital curve implies optimizing a set of objective functions in conflict (the considered error of the segmentation and the compression required in order to obtain that error) obtaining values for them which are acceptable to the decision maker (Osyczka, 1985). This definition leads to the question of who should play the decision maker role in a segmentation algorithm. Most presented approaches assign this role to the algorithm designer.

Consider the two different segmentations presented in figure 6.1. Both segmentations show different values for their objective functions, namely the error function and the number

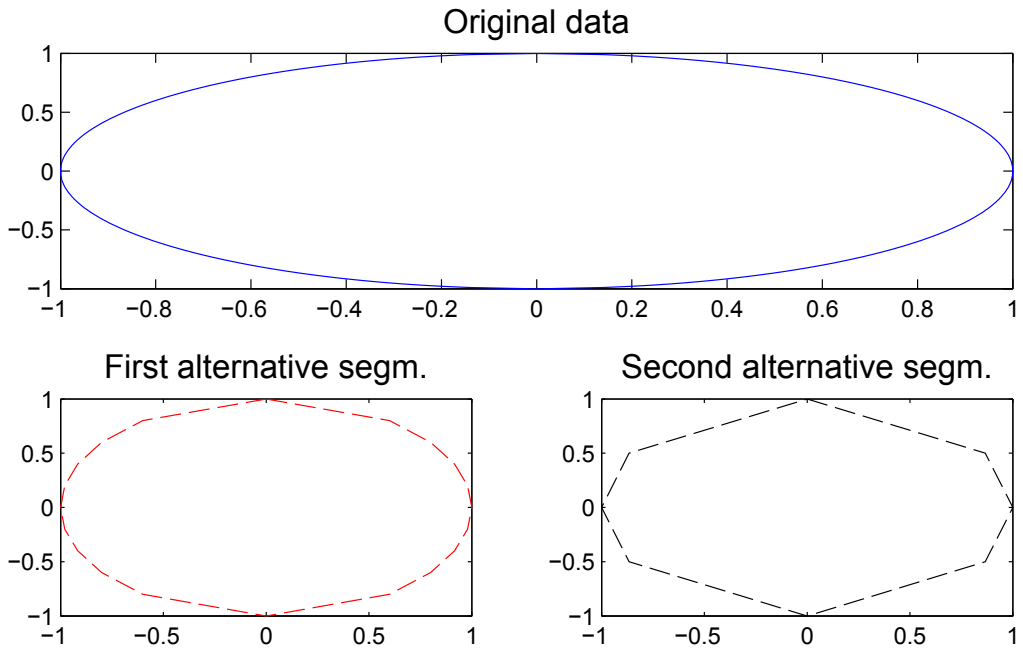


Figure 6.1: Alternative segmentations for a simple circle shape

of segments. The suitability of the representation depends on its particular application. Some may require a certain maximum error value, while others, due to their costly processing, may require a number of segments as low as possible. The range of possible processes is huge, from fast similarity search (Keogh et al., 2001) or data mining approaches (Keogh & Pazzani, 1998) up to optical character recognition applications (Pavlidis & Ali, 2007) or applications to the ATC domain as shown in chapter 3. Also, each of these processes may require different priorities for the different objective functions, and these requirements may change over time (e.g, different classifications may be preferred according to different available computer resources). This argumentation leads to the assignment of the decision maker role to the final user of the algorithm, considering as well that this DM may have changing preferences at different instants of time.

Available algorithms generally assume the algorithm designer to be also the DM, performing an a-priori dealing of the objectives in conflict, usually by means of an aggregating function (Surry et al., 1995). This implies that the algorithm designer establishes the importance of the different objectives and then codifies it into the algorithm running cycle. In other cases, the control over the secondary objective function may be implicit: as explained before, algorithms with a certain error tolerance as one of their input parameters may vary the compression value accordingly to that parameter value. This would imply that, for a scenario where the requirements of the decision maker (the final user) may change over time, the original data would have to be stored and the algorithm rerun with different parameters in order to deal with those different requirements. It is important to highlight that the choice of those parameters in order to meet certain requirements (especially regarding the implicit objective function values) can get to be very difficult to be performed accurately.

Multi-objective evolutionary algorithms (MOEAs) are evolutionary algorithms (EAs) focused on a set of different objective functions which have to be optimized jointly. The

objective of these algorithms is to find the optimal Pareto front (the set of solutions where improving an objective function value cannot be performed without degrading the value of a different one, (Ehrgott, 2005)). Evolutionary algorithms are a useful technique to deal with multi-objective problems, since they simultaneously deal with a set of possible solutions (the population) which allow them to find several members from the Pareto front in a single run (Coello & Lamont, 2004), instead of performing a series of separate runs, as had to be done with traditional mathematical techniques (Miettinen, 1999). They also have the interesting property of being less susceptible to the shape or continuity of that Pareto front (being able to deal with discontinuous and concave Pareto fronts).

The objective of this chapter is to propose a multi-objective solution based on genetic algorithms for the PLR segmentation problem to cope with the previous requirements: allowing the final user to decide from the best array of best found solutions considering the different objectives jointly (which will constitute the Pareto Front of the problem). The proposed approach eliminates the difficult *a-priori* parameter choices in order to satisfy the user restrictions (the solution choice is performed *a-posteriori*, from the obtained array of solutions) and allows the algorithm to be run a single time (since the whole Pareto front is obtained with a single run and different solutions may be chosen at different times from that Pareto front in order to satisfy different requirements).

The main contributions of this chapter are both theoretical and practical in their nature. First of all, the proposal and formalization of the segmentation issue as a multi-objective problem, along with the analysis of techniques available in the polygonal approximation literature regarding this multi-objective perspective and how it has been dealt with. This discussion includes the relevance of the decision maker role and who has been attributed this role in available approaches. This analysis leads to the proposal of an a-posteriori resolution method based on a standard MOEA , along with the required representation and operators (particularly focused on a specific initialization process regarding the nature of the objective functions). Finally, the proposed implementation is tested with its results comparison against a set of nine techniques from the polygonal approximation domain with a dataset of three standard curves, according to a single objective (quality of the individual elements of the obtained Pareto fronts compared to other techniques results) and multi-objective (measured by quality indicators) perspectives, highlighting the statistical significance of the obtained results.

6.2 Overview of segmentation techniques

One of the difficulties of detailing with the state of the art for the segmentation domain are the different naming conventions which similar algorithms receive in the different domains where they are applied (Keogh et al., 2003). A clear example of these different naming conventions may the Ramer algorithm (Ramer, 1972). That name is used in the image processing field, while in cartography is known as the Douglas Peucker algorithm (Douglas & Peucker, 1973), or the Iterative End-Point Fits algorithm, usually referred to in the machine learning community (Duda & Hart, 1973). Another commonly used name for this approach is the Top-Down algorithm (Keogh et al., 2003).

The objective of this section is to provide an insight into some different alternatives available in the segmentation domain to lead to the novel multi-objective metaheuristic proposal. This description of different algorithms will be used as the basis for the proposal

of the multi-objective technique presented in this work, and at the same there provide a considerable understanding of the approaches which have been taken to deal with the segmentation issue. For formalization purposes, we will start defining the components of the given time series with equation 6.1, where x_i and y_i are the plane coordinates of the point and t_i is the timestamp of the point's reception. If we are dealing with a closed curve without an explicit timestamp, that equation can be adapted following equation 6.2.

$$t = \{\vec{p_i}\}, \vec{p_i} = (x_i, y_i, t_i), i = 1, \dots, n \tag{6.1}$$

$$t = \{\vec{p_i}\}, \vec{p_i} = (x_i, y_i, i), i = 1, \dots, n \tag{6.2}$$

A general overview over segmentation techniques has been initially presented in section 2.8, including those usually applied o time series (section 2.8.2) and digitalization (section 2.8.3). This section is focused on digitalization techniques, covering them in a more complete way, such that specific analysis such as their handling of the multi-objective problem nature can be performed.

6.2.1 Teh and Chin algorithm

Teh and Chin algorithm (Teh & Chin, 2002) is based on the concept of the *region of support* (Langridge, 1972): this concept states that each boundary point of a closed curve must have its own view of the curve, being dominant points those which have a meaningful view of the curve which blocks the view of other non-dominant points.

In (Teh & Chin, 2002) the proposal is based on the difficulty of determining the curvature for a digital curve, which, in the real Euclidean plane, can be easily defined with equation 6.3. The functions to determine discrete curvature are named measures of significance (Rosenberg, 1972). Three different measures of significance are used: the k cosine measure, the k curvature measure and the *1* curvature measure. The k cosine measure was introduced in (Rosenfeld & Johnston, 1973) and is shown in equation 6.4. The k curvature measure was introduced in (Groen & Verbeek, 1978) and is shown in equation 6.5. Finally, the *1* curvature measure is derived from the previous measure (where $k = 1$), and is shown in equation 6.6.

$$\frac{\frac{d^2y}{dx^2}}{[1 + (\frac{dy}{dx})^2]^{3/2}} \tag{6.3}$$

$$cos_{ik} = \frac{\vec{a_{ik}} \cdot \vec{b_{ik}}}{|\vec{a_{ik}}||\vec{b_{ik}}|} \tag{6.4}$$

$$CURik = \frac{1}{k}\sum_{j=-k}^{-1} f_{i-j} - \frac{1}{k}\sum_{j=0}^{k-1} f_{i-j} \tag{6.5}$$

$$CURi1 = f_{i+1} - f_i \tag{6.6}$$

The algorithm starts with the calculation of the region of support for a given point p_i. This calculation is performed determining the length of the chord joining the points p_{i-k} and p_{i+k} (l_{ik}, shown in equation 6.7) and the perpendicular distance of the points contained in

the chord to their respective ones in the original data, d_{ik}. This process is continued until the value of the length of the chord stops growing or until the mean distance starts growing (represented in equation 6.8).

$$l_{ik} = \overline{|p_{i-k}p_{i+k}|} \tag{6.7}$$

$$\begin{cases} \frac{d_{ik}}{l_{ik}} \geq \frac{d_{i,k+1}}{l_{i,k+1}}, & \text{if } d_{ik} > 0 \\ \frac{d_{ik}}{l_{ik}} \leq \frac{d_{i,k+1}}{l_{i,k+1}}, & \text{if } d_{ik} < 0 \end{cases} \tag{6.8}$$

The second step of the algorithm calculated the three measures of significance represented in equations 6.4-6.6. Finally, according to the previous data, dominant points are calculated suppressing non-maximal points from the previous sets. This is performed with different procedures applied sequentially. The first one, suppresses the points, which, following the chosen $S(p_i)$ measure of significance, follow the condition in equation 6.9.

$$|S(p_i)| \geq |S(p_j)| \forall j \text{ such that } |i-j| \leq \frac{k_i}{2} \tag{6.9}$$

The second suppressing procedure eliminates those points with a value of zero in the *1* curvature measure. The third one analyzes adjacent surviving points, eliminating those with the lowest measure of significance. The process ends if the measure of significance is either the k cosine measure or the k curvature measure, being those points which have survived the previous filtering processes declared dominant.

However, if the chosen measure was the *1* curvature one, there is an additional final step which analyzes the groups of adjacent points which have survived the previous procedures. If a group has more than two points, the points at its edges are considered dominant points. If a group only has two points, the one with the highest measure value (or with the largest region of support, in case the both points have the same measure value) is considered dominant.

6.2.2 Marji and Siy algorithm

Marji and Siy algorithm (Marji & Siy, 2003) relies on the concept of *support arms*. This means that they do not use the region of support to calculate a significance measure of the boundary points, but instead compute the strength of the end points of their calculated regions of support, both in clockwise and counterclockwise directions. This strength is determined by the frequency of their choice. The idea is supported on an ideal corner shape, such as the one shown in figure 6.2, where the corner point would be chosen as an endpoint for all the different points in the shape, and thus, chosen as the dominant point.

To determine both support arms, the function shown in equation 6.10 is maximized, where L_{jk} is the length of the segment joining points p_j and p_k and E_{jk} is the sum of the squared perpendicular distances of the points contained between p_j and p_k to that segment. This is performed increasing iteratively the length of the region until that increase makes the function obtain a lower value. When that happens, the previous end point is considered the support point. k variable has an initial value of $j+2$ or $j-2$, depending on which support arm is being calculated.

$$F = L_{jk} - E_{jk} \tag{6.10}$$

Based on the previous concepts, the algorithm follows the following steps:

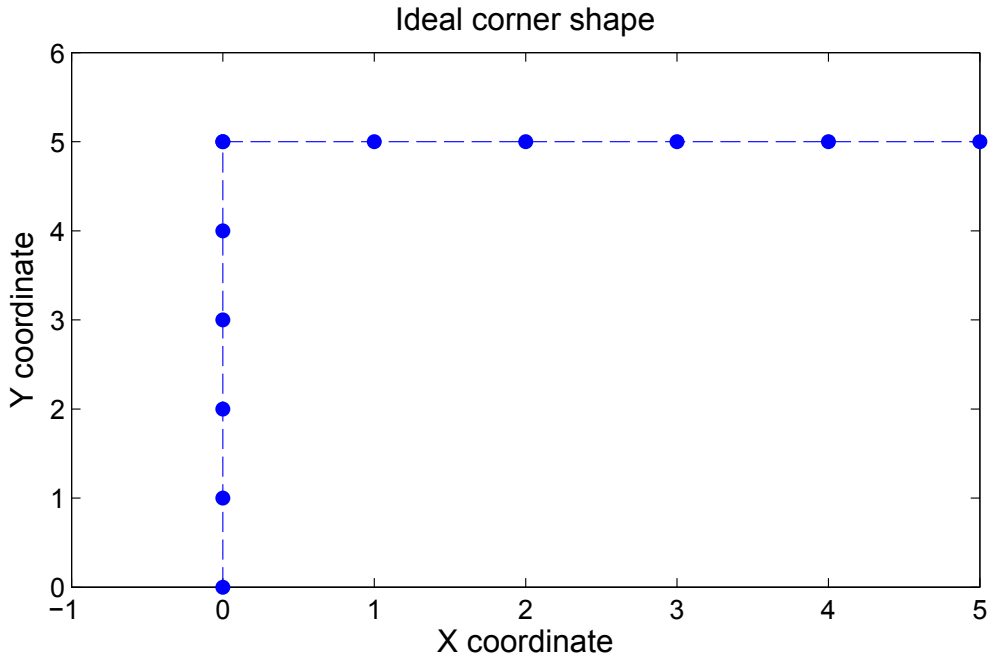


Figure 6.2: Ideal corner shape

1. All points are marked as non-dominant and uncovered

2. All points are analyzed to determine their support arms. If the considered node lies in an uncovered territory (adjacent points are uncovered) it is set as dominant, and the points in its regions of support are marked as covered. If it lies in a covered area, the perpendicular distance to the segment joining its closest dominant points both in clockwise and counterclockwise directions is calculated, being marked as dominant and covering its support regions if that distance exceeds 0.95

3. If a support end point is contained in the current region of support, the strongest points in the overlap segment are marked as candidate points. At the end of the iteration, candidate points are marked as dominant if their distance to the segment joining their closest dominant point in clockwise and counterclockwise directions exceeds 0.95.

4. If the point next to the support end point is also marked, if they have the same strength both are marked as dominant. In any other case, the strongest one is the only one masked as dominant.

6.2.3 Genetic approach based algorithms

Genetic algorithms have been used to deal with the polygonal approximation issue in a variety of ways (Goldberg et al., 1989; Pal et al., 1998; Tsai, 2006; Yin, 1999, 1998). These different approaches share many characteristics, such as the codification used, while they differ in specific choices, such as the crossover or mutation operators used. We will focus in this section in the Yin algorithm (Yin, 1999, 1998) and the speed-up modification introduced by (Tsai, 2006) to provide the required overview of the topic.

In Yin algorithm, from the formulation of the problem presented in equations 6.1 and 6.2, the codification proposed is a string of 1's and 0's as presented in equation 6.11, where $a_i = 1$ implies that a_i is a dominant point. The required fitness function of the genetic algorithm is expressed in equation 6.12, where R is a constant and $E(\alpha)$ is the approximation error between the segmentation result and the original data. Two different approximation error functions are proposed in the paper, the maximum error (E_∞, equation 6.13) and the integral square error ($E_2(\alpha)$, equation 6.14). In both cases, $e_i(\alpha)$ is the distance between p_i and the nearest line segment.

$$\alpha = a_1, a_2, \ldots, a_n \tag{6.11}$$

$$f(\alpha) = R - E(\alpha) \tag{6.12}$$

$$E_\infty(\alpha) = \max_{1 \le i \le n} e_i(\alpha) \tag{6.13}$$

$$E_2(\alpha) = \sum_{i=1}^{n} [e_i(\alpha)]^2 \tag{6.14}$$

In the algorithm the R value is adapted in order to prevent an outstanding individual to take a significant proportion of the following generation. To do so, they adopt the selectivity concept presented in (Singh et al., 1997): selectivity is the ratio of the maximally and minimally fit solutions in the population. Being E_{max} and E_{min} the maximal and minimal approximation errors for the individuals in the current population, the R value is calculated according to equation 6.15

$$R = \frac{E_{max} - E_{min}}{selectivity - 1} + E_{max} \tag{6.15}$$

The cross-over operator used is based on previous application specific approaches (Gen & Cheng, 1997; Pal et al., 1998), being specifically designed for their algorithm. The proposed operator forces the offspring to have the same number of dominant points as their parents, performed by swapping (0,1) pairs with (1,0) pairs appearing in the same position in the considered parents. The crossover probability, following (Loncaric & Dhawan, 1995), is a variable value adapted according to the generation value (based on the principle that the diversity of the population usually decreases with the increase of the generation number). It is calculated according to equation 6.16.

$$P_c = max(\sqrt{\frac{gen_{num}}{gen_{max}}}, 0.5) \tag{6.16}$$

The mutation operator performs a cyclic shift, in order to allow the number of dominant points to remain unchanged. This mutation operator is described in equation 6.17. The mutation probability is adaptative as well, following the same principles which led to the adaptative crossover probability (in order to increase the search space as the diversity is reduced with the increase in the generation number value). It is calculated according to equation 6.18.

$$a_{i \text{ modulo } n} = a_{i-1} \tag{6.17}$$

$$P_M = 0.3 * \frac{gen_{num}}{gen_{max}} \tag{6.18}$$

The algorithm uses an elitist strategy (Goldberg et al., 1989), where the fittest string in each generation is always taken to the following one. The rest of the genetic algorithm parameters are a population size of 100 and a number of generations of 100.

(Tsai, 2006) proposes several modifications over Yin algorithm, mainly to increase the speed required to obtain the solution. An additional table is added to the genetic algorithm, determining the probability of point p_i to be a *break point* regarding the current population. This probability is based on the k-cosine measure of significance (equation 6.4). The proposed probability function is shown in equation 6.19, where Z is the population size.

$$P_B(i) = \frac{\sum_{j=1}^{Z} cos_{ikj} + 1}{2Z} \tag{6.19}$$

The algorithm uses the same operators presented in Yin algorithm, but adds a divide-and-conquer technique based on the break point detection. Once a point has been determined to be a break point, the GA divides the chromosome in two parts according to the break point position and continues to be executed over both parts separately. The final solution is built upon the partial solutions of the different GAs built in this manner. Even though no result table is provided in the work, the graphical comparison included shows that similar results can be obtained with this technique in a smaller number of generations. The configuration parameters used are also different (a fact which does affect the number of generations required, even though no discussion was included in the work), setting the initial values of the population size to 60, the crossover probability to 0.6 and the mutation probability to 0.3.

It is remarkable that both algorithms require an input parameter: the number of segments in the solution. This fixed number of segments is the factor which creates the need for operators which do not alter the number of dominant points in the parents (if we are dealing with the crossover operator) or in the original individual (in the case of the mutation operator). Also, no way to guide the user in this choice is provided, being this a parameter which may be difficult to choose if the user does not have a-priori knowledge about the different qualities which different number of segments may provide over the curve approximation. This introduces the need to run the algorithms multiple times in order to determine which of those outputs meets the requirements of the solution. This need is shown in Yin's result presentation, where different results are provided regarding different possible number of segments configuration.

6.3 Multi-objetive approach to segmentation processes

The traditional criteria used in the data mining community to determine the quality of a segmentation process (Keogh et al., 2003; Liu et al., 2008), are the following:

1. Minimizing the overall representation error (*total_error*)
2. Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)

3. Minimizing the number of segments so that the total representation error does not exceed *total_error*

These criteria highlight the importance of the number of segments, but the comparisons performed, for instance, in the one of the source works for those criteria, (Keogh et al., 2003), are based only on the quality of the segmentation obtained, neglecting the cost of that quality. From the definition of the input data included in equations 6.1 and 6.2, we may formalize the definition of a segmentation process with equation 6.20, where each B_m would be the set of resultant segment, delimited by the dominant points at their extremes, k_{min} and k_{max}, and the number of those segments must be lower than n, the number of points in the original data.

$$S(t) = \{B_m\}, B_m = \{\vec{p_i}\}, i = k_{min}, \dots, k_{max}\ m \in [1, \dots, n-1] \tag{6.20}$$

Considering the previously stated criteria, we need to perform that segmentation according to a set of different objective functions which have to be minimized jointly, and which are in conflict. That problem matches perfectly the definition for a multi-objective optimization problem. The textual definition for these problems by (Osyczka, 1985) states that a *"multi-objective optimization problem can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions These functions form a mathematical description of performance criteria which are usually in conflict with each other Hence, the term optimize means finding such a solution which would give the values of all the objective functions acceptable to the decision maker"*. As seen in section 2.4.1, it may be formalized following equation 6.21.

$$\begin{aligned} f_p : \chi \to \Re,\ F(x) = (f_1(x), \dots, f_k(x))\ min_{x \in \Re} F(x) \\ \text{such that } \begin{cases} g_i(x) \leq 0 & i = [1 \dots n] \\ h_j(x) = 0 & j = [1 \dots m] \end{cases} \end{aligned} \tag{6.21}$$

Combining the segmentation problem formulation with the general multi-objective problem formulation according to the previous criteria, we obtain equation 6.22, which is the general formulation for the problem. In equation 3.3 $E(S(t), t)$, is the approximation error between the output segments of the process and the original data and $E(S(B_m), B_m)$ is the approximation error between the segment created by the dominant points of segment B_m (the edges of the segment) and the original points contained in B_m.

$$\begin{aligned} B_m = \vec{x_j}, j \in [k_{min}, \dots, k_{max}], m \in [1, \dots, p],\ p < n \to min\{E(S(t), t), p\} \\ \text{such that } \begin{cases} E(S(t), t) \leq total\_error \\ \forall m, E(S(B_m), B_m) \leq max\_segment\_error \end{cases} \end{aligned} \tag{6.22}$$

Once the problem has been formalized (this formalization had been initially required in chapter 3, to define the proper quality metrics), it is interesting to analyze the ways in which this multiobjective formulation has been tackled in the available algorithms. There are, basically, three different ways to deal with a multi-objective problem, (Coello et al., 2007). The definitions in the reference are restricted to multi-objective problems solved by means of evolutionary algorithms, but most of the definitions can be generalized to different approaches:

- A-priori techniques: These techniques require the DM, in general, to define the importance of the different objective functions in the MOP. The MOP is, with the use of these importance factors, reduced to a single objective optimization problem.

- Progressive techniques: These techniques require the direct interaction of the DM during the search process, combining cycles of search and decision making.

- A-posteriori techniques: A-posteriori techniques seek for P_{true} and PF_{true} (Horn, 1997), trying to perform a search as widespread as possible to generate as many elements as possible from the Pareto Set.

P_{true} is the *Pareto Optimal Set* and PF_{true} is the *Pareto Optimal Front*. The Pareto Optimal Set is the set of solutions where, changing their values, cannot improve one of the objective functions without degrading the value of another objective function. The Pareto Optimal Front is the set of objective function values associated to the Optimal Pareto Set. Their formal definition may be looked up in section 2.5.1. Applied to the segmentation issue, the Pareto Optimal Set would be the set of different segmentation solutions (each of them with a different number of dominant points) where changing the number of dominant points in any of those solutions would result in a solution with a worse approximation error than one of the solutions already included in the Pareto Set. This means that the output for a segmentation process seeking that Pareto Optimal Set would be the best possible segmentation solutions with different compression levels (being a compression level the rate between the original points in the curve and the dominant points in that particular element of the Pareto Set).

The different techniques presented in section 6.2 deal with the problem according to a-priori techniques. This means that they turn, with different mechanisms, the multi-objective problem into a single objective problem, and optimize that single objective problem with their particular techniques. Different a-priori techniques include lexicographic ordering (Fourman, 1985), aggregation functions (Surry et al., 1995) or converting objective functions to input parameters. Lexicographic ordering imposes an order among the different objective functions, and the best fitted individual is obtained according to the most important objective function, using the others as secondary fitness values to solve tie situations. Aggregation functions build a single fitness value combining the different objective function values. Finally, converting an objecting function into an input parameter focuses the search of the algorithm into a single element of the Pareto Set, leaving the DM with the responsibility of determining the rest of the characteristics of that element.

Teh and Chin algorithm (section 6.2.1) uses both aggregation functions and lexicographic ordering techniques. Aggregation functions are used at different steps: computing the region of support, it continues to grow while the mean distance value does not increase. That mean distance value (equation 6.8) is an aggregation function, using the length of the segments and the approximation error. Also, the suppression condition in equation 6.9, uses a combination of different objective functions (the measure of significance and the length of the region of support) for its decision. Finally, the suppression process performed as a final step when the *1 curvature* measure of significance was chosen, uses lexicographic ordering to determine which is the dominant point in surviving groups with only two points, using the measure of significance as the priority objective function and the size of the region of support as the secondary objective function.

Marji and Siy algorithm (section 6.2.2) uses aggregation functions both explicitly and implicitly. The function to determine the length of a supporting arm (equation 6.10) is an aggregation function using again the length of the support arm and the approximation error as the combined objective functions. Also, the process to determine whether a candidate point must be considered a dominant point or not, chooses a non-explicit aggregation function, since choosing it as a dominant point would reduce the length of the segments on the output, and that choice is taken according to a threshold over the approximation error.

The presented evolutionary techniques (section 6.2.3) deal with the multi-objective nature of the problem converting the *number_of_segments* objective function into an input parameter determined by the user. This choice can be analyzed from two opposite points of view: if the user knows which is the compression level he requires for his application, this allows the calculation of the best solution focused only on that compression ratio. This idea can be implemented to perform automated batch processing of data sets according to the multiplication of the compression ratio by the number of measures in the time series. However, the results obtained for the error may not be feasible for the application of the results, leading to the need of individual choices for the number of segments in each input time series, and requiring the constant feedback from the DM during the whole process. The use of constrains in the evolutionary approaches might be a solution to deal with this issue, but the choice of those constrains would be individual for each input. In (Yin, 1999, 1998) these difficulties are met providing different solutions for different *number_of_segments* parameter values. Each of these solutions runs the evolutionary algorithm from an initial random population.

This requirement for different possible solutions is not only met in evolutionary techniques. Traditionally, Pareto fronts were built by mathematical techniques for multi-objective optimization artificially by performing several runs with different parameters(Miettinen, 1999). In non-evolutionary techniques for segmentation purposes, input parameters are commonly based on the approximation error rather than the number of segments, being also a representative amount of non-parametric techniques (which obviously can never produce a Pareto front, since they can only provide a single solution for each problem instance). In parametric techniques, in order to build a complete Pareto front, the user must determine the approximation errors to obtain the required number of segments in the approximations. The choice of these values may be an optimization issue itself, and clearly problem dependent. This implies that, in the cases where such a solution is possible (parametric techniques) it is difficult and computationally costly to obtain a Pareto front for a segmentation problem with the available approaches.

6.4 Multi-objective evolutionary algorithm for segmentation processes

A-priori techniques have a series of difficulties in their treatment of MOOPs: difficulties regarding continuity and shape, the need to be run several times to obtain several individuals from the Pareto front... In the segmentation domain, the difficulties regarding the configuration of the techniques to obtain the different elements of that Pareto front must also be taken into account (the choice of the input parameters values in order to obtain certain values in the objective functions is clearly not trivial, such as determining a certain maximum

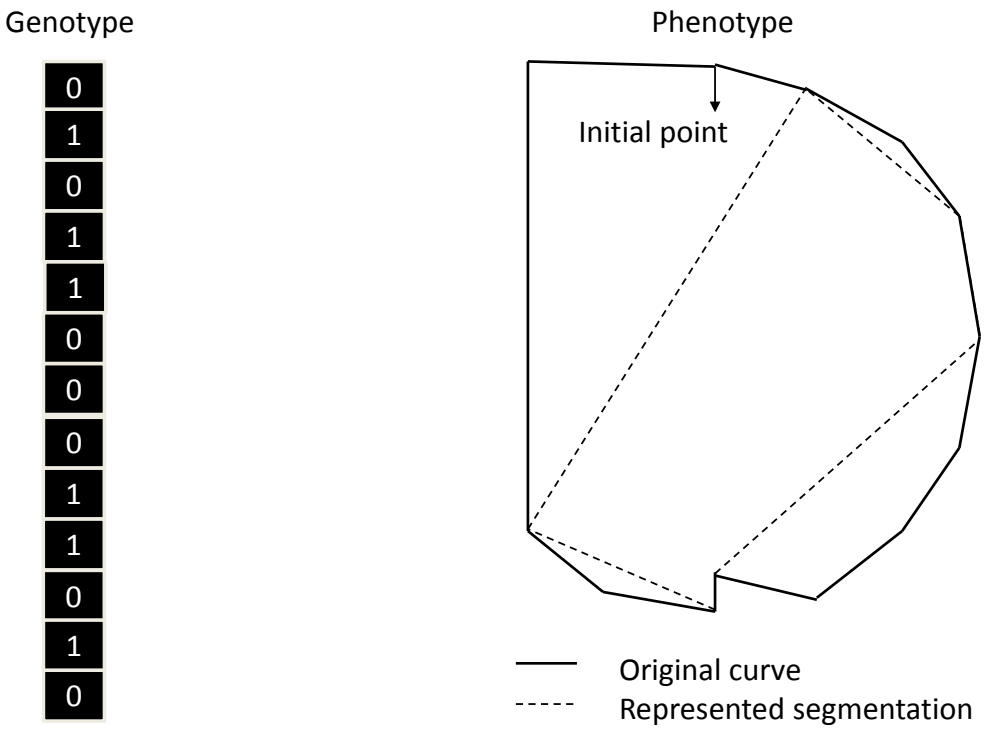


Figure 6.3: Genotype to phenotype mapping

error in the approximation segments in order to obtain a segmentation solution with a certain number of dominant points).

The purpose of this work is to deal with the segmentation issue with the use of a *a-posteriori* technique, according to the formulation presented in equation 6.22.

6.4.1 Representation

The first issue regarding representation of the problem is the choice of the related structure. In traditional approaches, such as the one covered in section 6.2.3, the representation was based on the detection of dominant points, codifying each problem instance as a string of 0's and 1's, representing each gene a point in the problem instance and whether this was a dominant point or not. Figure 6.3 shows the relationship between the genotype and its represented phenotype.

An alternative possible representation can be based on integer values, representing each of these integer values the position of the point in the input problem instance. This representation could be based on a fixed or variable size chromosome. The chosen alternative could be a fixed size chromosome where this size was equal to the input problem instance size (such as in the previous approach), such that dominant point might be repeated in that structure. Figure 6.4 shows an example of this approach.

This representation attempts to provide a representation anchor to the importance of certain key dominant points, which are present in almost all the different possible segmentations, regardless of the number of dominant points used (this can be seen in figures 6.11-6.13).

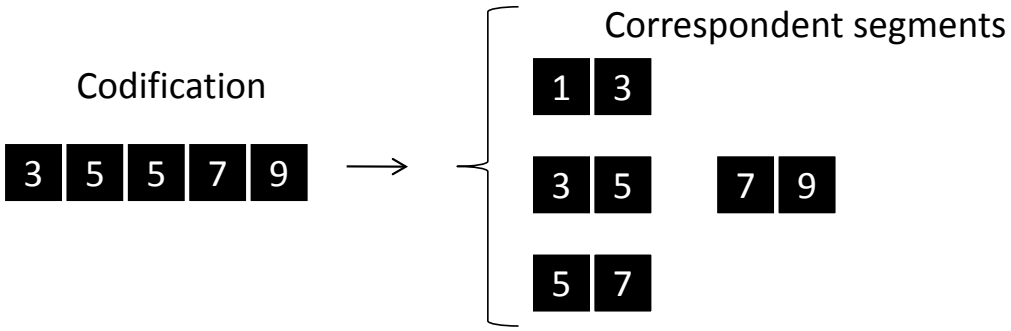


Figure 6.4: Integer representation genotype

By storing several copies of those important dominant points in a chromosome, they would become more resilient to the changes introduced by transformation operators. Also, it introduces a series of handicaps: first of all, the chromosome has to be ordered in order to provide efficient transformation operators, and reordering has to be applied after the application of any transformation operator, affecting the performance of the algorithm. An even more important handicap is related to the fact that the search space is much more extensive than the one obtained using a binary representation. Also, there is no direct genotype to phenotype relationship, since now several different genotypes can represent the same phenotype. This fact may make the search slower and affect the efficiency of transformation operators.

Early experiments were conducted regarding the representation choice, where the binary codification seemed to be more promising, mainly due to its reduced search space. Results for the integer representation were tested versus some of the traditional techniques presented in section 3.3, focusing on the one reported to be more accurate, the bottom-up algorithm. The results, regarding the obtained Pareto fronts for two of the figures in the dataset presented in section 3.6.2, are presented in figures 6.5 and 6.6.

The presented formulation introduces some restrictions which may help to reduce the search space, but the choice of those boundaries may be problem dependent and also (according to previous argumentations), not trivial to establish. Thus, the segmentation issue will be faced as a multi-objective problem without restrictions, obtaining as many elements as possible from the *Pareto Set* and *Pareto Front* and letting the DM choose from those final solutions. An important consideration is that this choice is made from final solutions with all their characteristics, rather than a priori configuration values which may lead to unfeasible results in other components of the solution vector, allowing the DM to make simple choices (and also to vary them according to different needs for different processes, as was discussed in the introduction section).

6.4.2 Initialization

Convergence speed is a constant issue in evolutionary computation, and it has been approached with modifications in the different involved processes: crossover, mutation, selection, etc. Initialization procedures have received a reduced amount of interest from the research community, generally assuming that the overall impact over the performance of the algorithm is lower. Many genetic algorithms use a default bitstring uniform initialization pro-

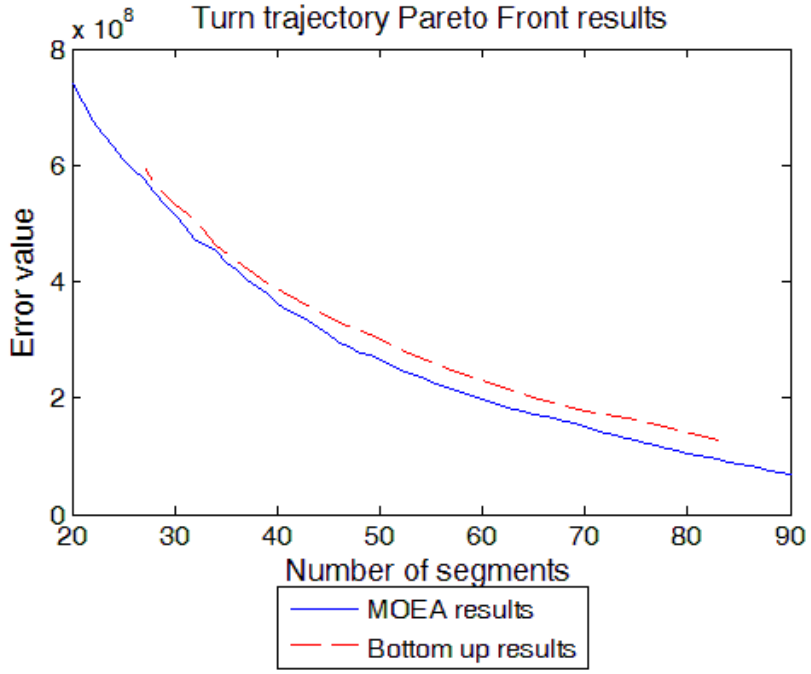


Figure 6.5: Integer representation results for turn problem instance

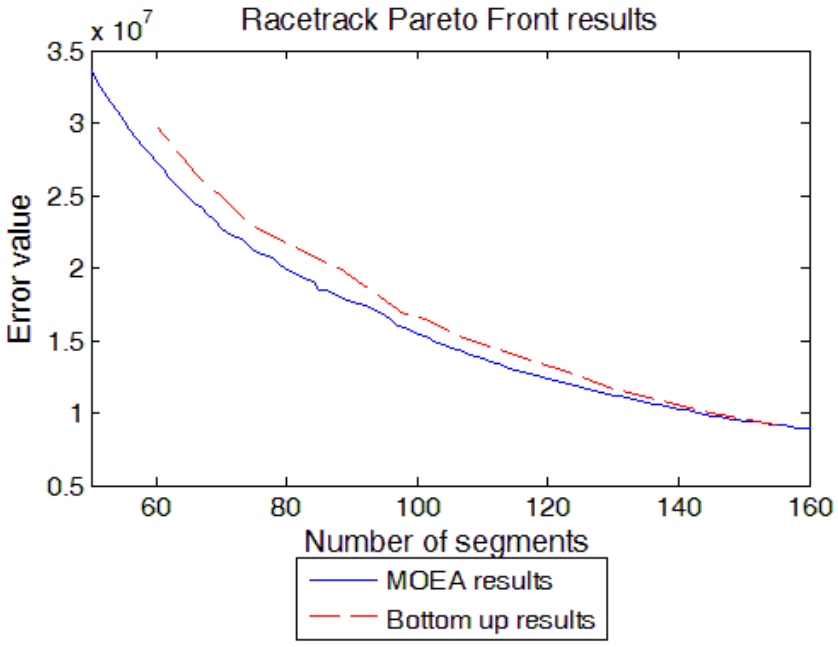


Figure 6.6: Integer representation results for racetrack problem instance

cedure, assigning values of 0 or 1 to every bit for each individual in the population, obtaining a uniform population regarding the binary space, which also exhibits the maximal bit-wise diversity (Kallel & Schoenauer, 1997). However, early research showed that this may not be the optimal initialization procedure for specific domains, such as inverse problems in Structural Mechanics (Schoenauer, 1996), where the solutions were known to contain far more 0's than 1's.

General approaches have to provide a trade-off between the improved initial population obtained and the cost of the process. Such a discussion is carried out in (Rahnamayan et al., 2007), where opposition-based and quasi-random (Maaranen et al., 2004) initialization methods are compared, highlighting the computational issues and dimensionality effectiveness. In (Ramsey & Grefenstette, 1993) the reuse of previous solutions in terms of population initialization is considered for the application of evolutionary algorithms to dynamic environments, but the established principles can be used for static environments where an approximation to the solution is known (or can be calculated, as in the local search based method compared in this work). Finally, domain specific approaches introduce characteristics from the faced problem in order to include a seeding in the initial population which can improve the overall results. In (Burke et al., 1998) such an approach is studied for the timetabling problem, where heuristic individuals go through some randomization process in order to generate the initial population, presenting a discussion of the diversity effect of such a process over the final outcome of the algorithm.

Three different initialization procedures will be compared for the presented problem: default (bitstring uniform), uniform (in terms of Pareto front) and local search. Default initialization assigns a 50% chance of becoming a *dominant point* to each point in the original data. According to that probability, this method generates an initial population which, in the number of segments objective function, is centered around 1/2 of the number of original elements in the data. Being this objective also closely related to the representation error, this generates a poor diversity on the number of segments (or, similarly, the number dominant points), which also implies a poor diversity on the covered range of approximation errors.

Even though default initialization produces the maximal bit-wise diversity, a poor one is obtained in the resultant Pareto front. Since multiobjective optimization seeks the Optimal Pareto Set in the variable space and its associated Optimal Pareto Front in the objective space (the set of solutions where one solution objective function value cannot be improved unless another objection function value is degraded (Coello et al., 2007)), this may not be the optimal strategy. Uniform initialization tries to ensure the diversity of the front obtained. To achieve this task, each individual is generated according to a number of random dominant points, which are then included into the chromosome at random gene positions. This generates a population which is spread along the dominant points objective, obtaining as well a good diversity over the representation error objective function. Related to the initialization approaches presented at the beginning of this section, this approach is general (in terms of exploiting the Pareto front diversity in the initial population) but uses a domain specific procedure to produce the front with a very low computational cost.

Local search initialization is a heuristic seeding approach using bottom-up segmentation (Keogh et al., 2003) to introduce good individuals into the initial population, a technique which is claimed to obtain comparatively better results than other offline alternatives. This algorithm creates the finest possible approximation of the time series, dividing it into n-1 (where n is the number of points in the time series) segments of length value 2. Afterwards,

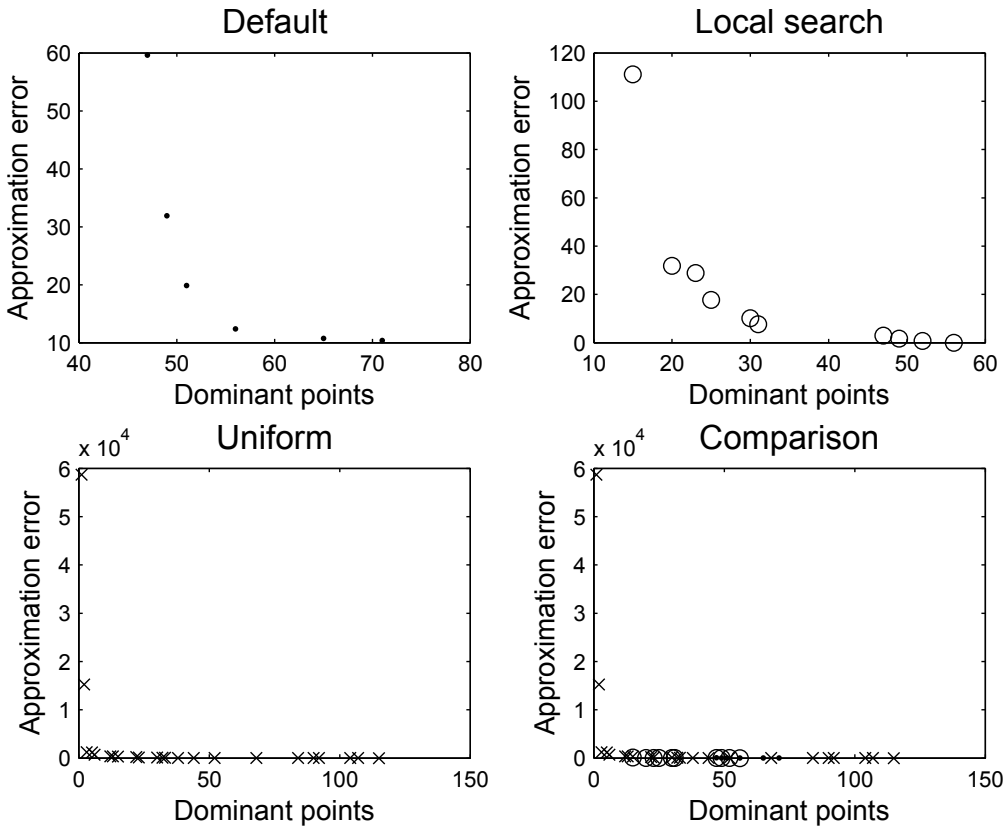


Figure 6.7: Initial Pareto front comparison for the three presented methods (leaf curve)

the cost of merging each pair of adjacent segments is calculated and, if the merge with the lowest cost has an error bellow the user defined value, the segments are merged. The process continues until no pair of adjacent segments can be merged with an acceptable error value. It is important to notice that in every step of the algorithm the costs of the adjacent segments to the merged one in the previous step must be updated.

One of the difficulties arising in the application of these single objective procedures is that, in order to obtain a certain number of different individuals to be introduced into the initial population, there is a lack of direct control over the objective functions values. This may require several executions to obtain a single individual which can be introduced into the population, thus increasing the overall computational cost. On the other hand, unlike other presented alternatives in the literature (Burke et al., 1998) different individuals are obtained with the different configuration parameters directly from the heuristic technique, eliminating the requirement for additional randomization processes.

Figure 6.7 presents the non-dominated solutions obtained in an initial population of 100 individuals generated with the default method and the proposed approach based on the diversity in the objective space, along with a Pareto front composed from ten solutions obtained with different runs of the detailed single-objective algorithm. As expected, the range in the objective functions covered by the default initialization is very limited compared to the one which focuses on objective function diversity. The number of non-dominated individuals generated is clearly inferior to those in the uniform approach as well, obtaining

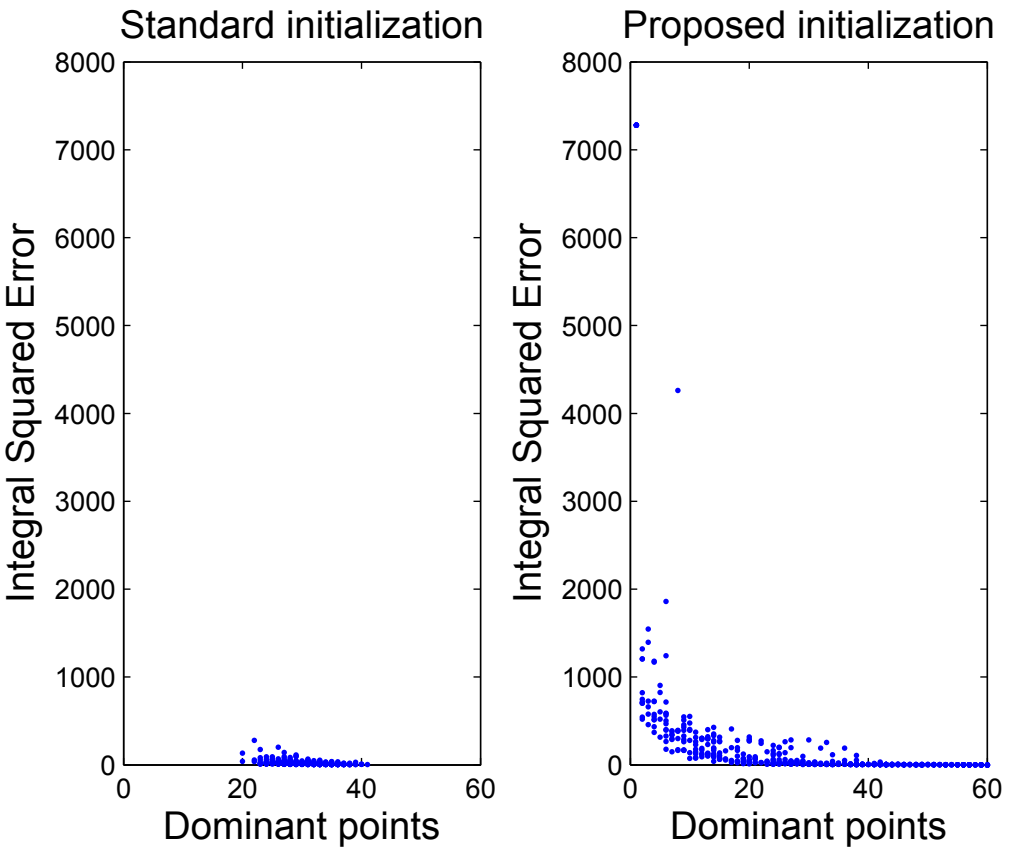


Figure 6.8: Initialization processes comparison

an initial population which, even though it is composed of the same number of individuals, provides the algorithm with less valuable information (Pareto front individuals). Local search initialization provides individuals which are clearly superior to the ones randomly initialized (by either of the alternative procedures), but their range is limited compared to the ones performed by uniform initialization.

6.4.3 Underlying MOEA algorithm: SPEA2

The focus of this chapter is not to proof the benefits of a particular technique (even though one has been chosen for the results presentation and comparison), but rather of the whole approach itself. To do so, we will choose a very extended MOEA: Strength Pareto Evolutionary Algorithm 2 (SPEA-2) (Zitzler et al., 2001), according to its implementation in the JMetal integrated development environment (IDE) (Durillo & Nebro, 2011). The choice of this algorithm has been made according to its extended implementations in different languages and IDE's which can ease the comparison with the results presented for different authors, along with its wide usage in research works. Also, it was chosen over alternative algorithms which share similar wide usage characteristics (such as NSGA-II, (Deb et al., 2002a)) due to its use of an *archive* to preserve the best solutions among different generations, which suits the requirements of segmentation algorithms.

The main characteristic of the Strength Pareto Evolutionary (SPEA) algorithms is the concept of *strength*, which defines their name. These algorithms define an external archive, where non-dominated solutions are saved (called external nondominated set). This archive is updated with nondominated individuals after each generation is processed, and the *strength* value of each of these individuals is computed. The computation of this value was originally proportional to the number of solutions which a certain individual dominates (in the original SPEA algorithm) but in SPEA2 it was changed to a value which depends on both, the number of individuals which a certain individuals dominates and the number of individuals which dominate it.

The environmental selection determines how this archive is updated. Originally, a clustering technique was used, but this process tended to lose boundary solutions when the size of the archive was too small for the required number of nondominated solutions. An enhanced truncation technique is present in SPEA2, which invokes an iterative process that eliminates at each stage the individual with the minimum distance to another individual (in case of ties, the distance to the second closest individual is considered and so on) until the number of nondominated individuals in the archive fit its maximum size. This process allows the algorithm to retain the boundary solutions through its different generations. The size of the archive in the SPEA2 algorithm remains constant, implying that if the number of nondominated solutions at a certain generation is less than the archive size, this archive is filled up to its size with dominated solutions.

The computational complexity of the algorithm is dominated by the environmental selection procedure, with a worst case complexity for the truncation operator of $O(M^3)$, where M is the population size plus the archive size. On average, that complexity is reduced to $O(M^2 logM)$, which is also the complexity presented by the fitness assignment process. In the application case presented in this work, the established configuration parameters, which will be detailed afterwards, guarantee that the truncation operator will not be required, setting the overall algorithm complexity to this reduced form. The pseudo-algorithm of SPEA2 can be defined as follows:

1. Initialization: Generate initial population P_0 and create the initial empty archive $\overline{P}_0$
2. Fitness assignment: Calculate fitness values for individuals in P_t and $\overline{P}_t$
3. Environmental selection: Update $\overline{P}_{t+1}$ according to the explained procedure
4. Termination: Check stopping criteria. If it is met, output nondominated individuals in $\overline{P}_{t+1}$, otherwise continue.
5. Mating selection: Binary tournament selection with replacement on $\overline{P}_{t+1}$ to fill the mating pool.
6. Variation: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Go to step 2.

The use of the multi-objective approach has been presented from the perspective of the problem formulation and also as a mechanism to prevent some of the difficulties found in available methods in the literature. Once an evolutionary algorithm has been chosen for this task, we may also determine whether the MOEA approach is suitable to handle this issue. The key to this approach is the fact that the different solutions of the Pareto Front can share

valuable information among them. This can be seen in figures 6.11-6.13, where different approximations with a different number of dominant points share similar positions of their key dominant points, and increasing their number leads to a more refined approximation of the zones with more abrupt changes. This implies that the use of the MOEA for this purpose also balances the high computational cost of using an evolutionary algorithm with the advantage of obtaining the whole set of solutions of the Pareto Front at a speed much faster than the time required to obtain them individually.

6.4.4 Technique configuration

The configuration required for the chosen technique implies the mutation and crossover probabilities, population size and number of generations (the rest of the parameters are chosen according to their standard values: 1-point crossover, bit-flip mutation and binary tournament selection). The first two probabilities have been chosen according to standard values (0.9 for the crossover probability and $1/chromosome\_length$ for the mutation one). Population size and number of generations did not have a clear choice, so a set of experiments was run with population sizes ranging from 100 to 500 and generation values ranging from 100 to 2000 in order to determine their values. In order to establish whether there were significant improvements between the different configurations, a Wilcoxon test (Corder & Foreman, 2009) was used over the hypervolume result (Zitzler et al., 2003) of the obtained Pareto Fronts, with 30 runs for each configuration over the three curves in the used dataset. In table 6.1 the results for this comparison over the chromosome curve are shown, where 0 means that there is no statistical significance at 1% level, 1 means that there is statistical significance and "-" that the comparison is not applicable or already covered. The configuration values for each configuration number with a population size of 100 are shown in table 6.2. Configuration numbers 7-12 share the same growing generation values with population size 200, and configuration numbers 13-18 with population size 500.

Last column of table 6.1 shows that there is no statistical significance in the difference of the presented results between the runs with population size 500 and 1000/2000 generations (doubling the computational efforts does not provide additional improvements over the quality obtained). If similar tests are run over the other two figures in the dataset, that increase over the run generations value does provide improvements over the results, so, in order to set the same configuration parameters for the three curves in the dataset, we will use a population size of 500 and generation number of 2000.

The summary of the proposal is presented in table 6.3.

6.5 Experimental results

The dataset used will be based on the three most extended curves for polygonal approximation testing, usually named chromosome (figure 6.11), leaf (figure 6.12) and semicircle (figure 6.13). We will compare the results obtained with a set of nine representative techniques, some of which have been detailed in previous sections: (Marji & Siy, 2003), (Teh & Chin, 2002), (Sarfraz et al., 2004), (Cronin, 1999), (Ansari & Huang, 1991), (Ray & Ray, 1992), (Sarkar, 1993), (Wu, 2003) and finally a special comparison with the evolutionary technique by (Yin, 1999). Before this comparison is performed, we will compare the different initialization techniques presented in section 6.4.2, choose one among them, and, setting this initialization

Table 6.1: Wilcoxon test results for different MOEA configurations applied to the Chromosome curve

Conf.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
2	-	-	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
3	-	-	-	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
4	-	-	-	-	0	1	1	0	0	1	1	1	0	1	1	1	1	1
5	-	-	-	-	-	0	1	1	0	0	0	1	0	0	1	1	1	1
6	-	-	-	-	-	-	1	1	1	0	0	1	1	0	0	0	1	1
7	-	-	-	-	-	-	-	1	1	1	1	1	1	1	1	1	1	1
8	-	-	-	-	-	-	-	-	0	1	1	1	0	1	1	1	1	1
9	-	-	-	-	-	-	-	-	-	0	1	1	0	0	1	1	1	1
10	-	-	-	-	-	-	-	-	-	-	0	1	0	0	1	1	1	1
11	-	-	-	-	-	-	-	-	-	-	-	1	1	0	0	1	1	1
12	-	-	-	-	-	-	-	-	-	-	-	-	1	1	0	0	0	1
13	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	1	1	1
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1	1	1
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	1
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 6.2: MOEA configurations detail for population size 100

Config. number	1	2	3	4	5	6
Population Size	100	100	100	100	100	100
Generations	100	300	500	700	1000	2000

Table 6.3: Multi-objective segmentation algorithm summary

Parameter	Description
MOEA algorithm	SPEA2
Representation	Binary vector (0= nondominant, 1=dominant)
Objective Functions	2, dominant points and Integral Squared Error
Initialization process	Problem specific
Crossover operator	1-point crossover
Mutation operator	bit-flip mutation
Crossover probability	0.9
Mutation probability	$1/n$
Population size	500
Generation number	2000
Archive size	$n-1$

Table 6.4: Initial populations comparison

Chromosome curve								
Default		L S	Uniform		Unif +l s		Def +l s	
Mean	Std	Mean	Mean	Std	Mean	Std	Mean	Std
4 47E-01	4 39E-02	8 59E-01	9 54E-01	7 52E-03	9 60E-01	6 82E-03	8 59E-01	9 94E-07
Leaf curve								
1 66E-01	3 22E-02	7 45E-01	9 62E-01	1 99E-02	9 63E-01	1 99E-02	7 45E-01	3 39E-16
Semicircle curve								
2 80E-01	5 21E-02	8 08E-01	9 50E-01	2 42E-02	9 51E-01	2 42E-02	8 08E-01	4 52E-16

procedure for the final algorithm configuration, perform the comparison with the chosen techniques from the domain.

6.5.1 Initialization results

Along with the performance of the presented initialization methods, current experimental validation will try to determine whether the inclusion of local search individuals in the population generated by either of the alternative methods improves its results. For the validation of the performance of the different initialization methods, 30 runs of every configuration have been performed, the unary hypervolume (Zitzler et al., 2003) of the resultant Pareto Front calculated for each of the alternatives (both for the initial and final populations), and the difference between the different pairings calculated. Afterwards, a t-test is carried out to determine the statistical significance of the obtained results. The reference front used for the hypervolume computation is obtained with a uniform initialization procedure and a population size of 1000 individuals left to run for 2000 generations, such as established in section 6.4.4.

The representation of the initial population Pareto fronts for the three curves in the dataset are presented in figures 6.7 (leaf), 6.9 (chromosome) and 6.10 (semicircle). Graphically these figures show several interesting facts regarding the proposed initialization: assuming that the heuristic seeding provided by the local search technique provides good solutions in terms of objective functions values and diversity, the comparison with the default process shows that bitstring uniform populations may provide good (figure 6.9) or very bad solutions (figure 6.7), being this quality problem dependent (determined by whether the solutions around 50% dominnant points are meaningful or not for the final Pareto front), disencouraging the use of this technique for an unknown problem instance. On the other hand, the initial populations provided by the uniform method exhibit for all the different dataset instances Pareto fronts with a very good diversity over the two objectives, being thus applicable to new unknown instances with a certain guarantee over the quality of the initial population's Pareto front.

The hypervolume results obtained for the three different curves are presented in tables 6.4 and 6.5, while the statistical significance results over those values are presented in table 6.6. The initial populations comparison does not provide a standard deviation value for the local search initialization, since each of the runs starts with the exact same initial population. In final populations, no results for local search are provided, since, as will be detailed, the populations obtained by local search dominate those created by a default initialization process, providing the same final results (disregarding the stochastic nature of evolutionary approaches) in local search and local search plus default initialization configurations (being

Table 6.5: Final populations comparison

Chromosome curve								
Default		Uniform		Unif +l s		Def +l s		
Mean	Std	Mean	Std	Mean	Std	Mean	Std	
9 41E-01	2 97E-02	9 67E-01	1 07E-04	9 66E-01	4 70E-03	9 67E-01	1 07E-04	
Leaf curve								
7 77E-01	4 76E-02	9 79E-01	3 55E-03	9 76E-01	1 20E-02	9 77E-01	7 58E-03	
Semicircle curve								
8 46E-01	4 06E-02	9 75E-01	5 27E-03	9 76E-01	3 08E-03	9 77E-01	3 39E-04	

Table 6.6: Statistical significance test

Curve	Def/l.s.	Def./Unif.	Unif./l.s.	Unif./Unif. + l.s.	Def./Def. + l.s.
Chromosome	No	Yes	Yes	No	No
Leaf	Yes	Yes	Yes	No	Yes
Semicircle	Yes	Yes	Yes	No	Yes

these results included under this last heading).

The test results presented in table 6.6 are obtained from the final populations, since all the differences in the initial ones were statistically significant. The results show that uniform initialization yields better performance of the algorithm compared to any of the remaining alternatives, and also that the addition of local search individuals to its initial population does not improve its results (in the final outcome of the algorithm). However, local search use does improve (for the two harder problem instances, lead and semicircle) the default initialization performance.

The initial populations provided by the different runs of a default initialization procedure become, in general, fully dominated by the individuals introduced by the local search (results in table 6.4 for local search and local search plus default individuals are the same). The impact of the local search procedures is related to the quality of its results compared to the optimal Pareto front and the cost of their computation. As presented in table 6.6, the heuristic seeding does improve the results of the bitstring random initialization process (in two of the three curves in the dataset), but also requires a computational cost to obtain those individuals. As previously explained, obtaining n individuals for this initial population by means of the local search procedure may require more than n executions of this algorithm, and this cost may be even higher if certain diversity is required in those heuristic individuals.

Uniform initialization provides a higher range of objective function values to its individuals (which are graphically represented by the initial and final *"tails"* of the Pareto front), which provides additional non-dominated individuals to the algorithm and allowing it to obtain better final solutions, as seen in table 6.6. This shows the importance of the diversity in terms of objective space, which cannot be obtained with the default bitstring random initialization. Even though there is no general technique to be able to obtain this diversity in the objective space for a general problem, the presented technique allows to do so in the segmentation domain with a very low computational cost (similar to that of the default initialization process)

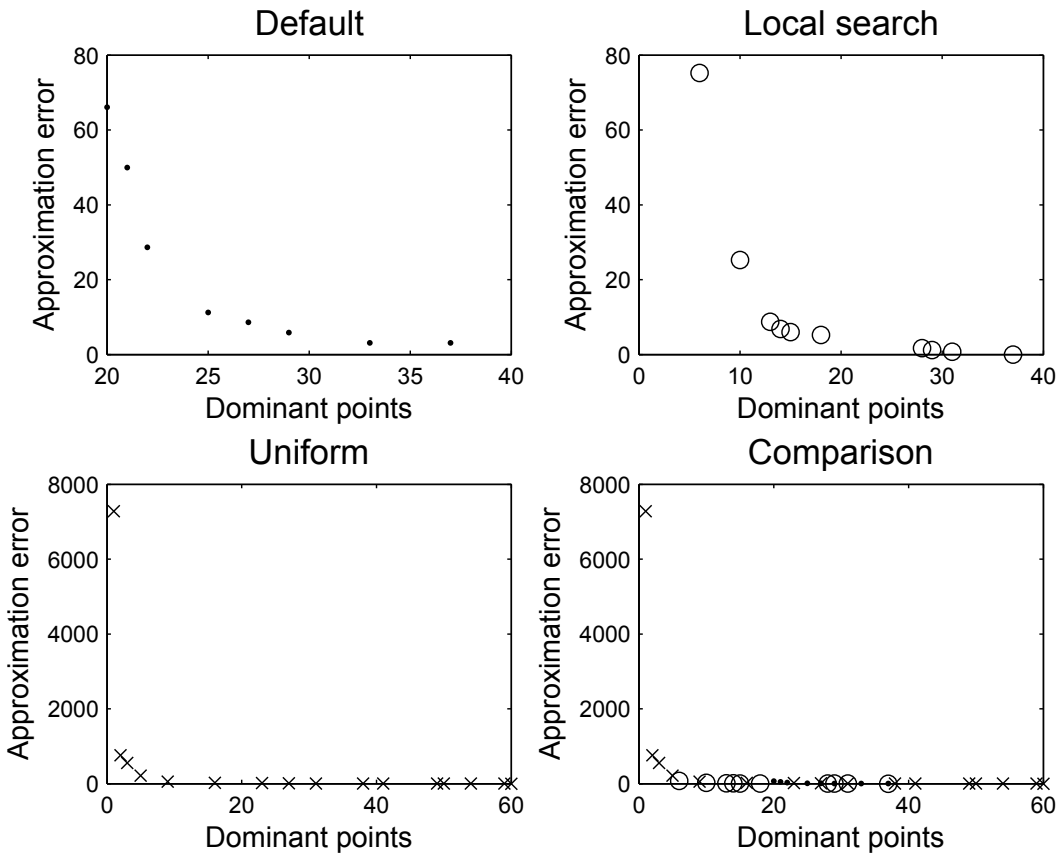


Figure 6.9: Initial Pareto front comparison for the chromosome curve

being clearly superior to the considered alternatives. Therefore, for the remaining comparison experiments, the uniform initialization will be the procedure used.

6.5.2 General comparison results

The dataset, along with some segmentation results from the obtained Pareto Fronts, is presented in figures 6.11-6.13. Figure 6.11 introduces the chromosome curve, which has 60 boundary points, along with five results from the Pareto Set obtained by the technique. Figure 6.12 shows the same results for the leaf curve (with 120 boundary points), and figure 6.13 for the semicircle one (with 102 boundary points). The numerical description of these figures, according to their freeman chain code (Freeman, 1961), is presented in table 6.7.

Table 6.8 presents the concrete results for the Pareto fronts obtained with the presented configuration of the technique for the three curves of the dataset, showing the number of dominant points in the element and the integral squared error of that element. There are several interesting facts in those results: first of all, the technique is able to find the number of segments which produces a lossless approximation over the different curves. Secondly, the leaf curve Pareto front approximation results have no value for 55 dominant points, while it reaches its lossless approximation with 56 dominant points. This does not mean that the algorithm was not able to find a solution with 55 d.p., but rather that the solution found

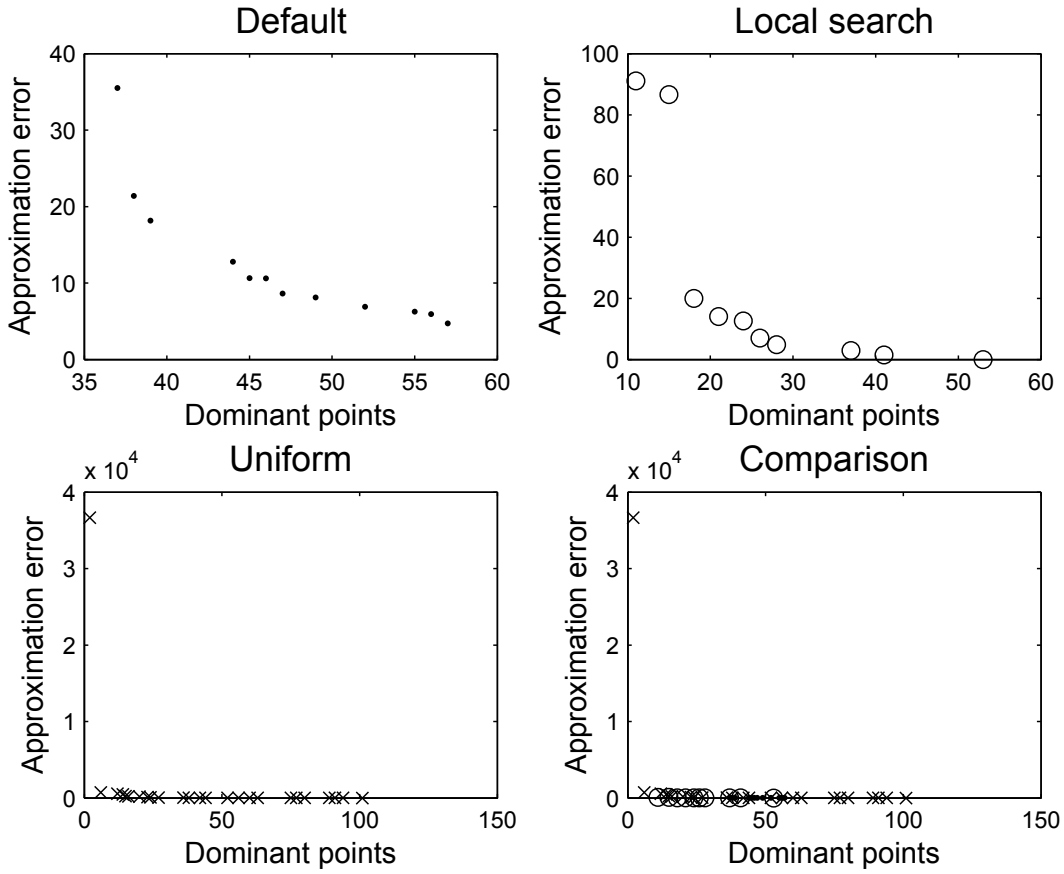


Figure 6.10: Initial Pareto front comparison for the semicircle curve

did not improve those found with a lower number of them, and according to that (it is a dominated solution according to Pareto dominance) it was removed from the Pareto front. Mean and standard values corresponding to 30 different executions are provided for all the different elements in the dataset.

Table 6.9 presents the results of the first eight techniques to be compared. These technique results are either non-parametric or the included results are those presented in their reference works according to their default configuration. This means that each of these techniques provides only a single solution for each problem in the dataset. Table 6.10 presents the statistical comparison of these techniques with the MOEA technique used. To perform this comparison, the solution with the appropriate number of dominant points (the same as the single solution provided by the compared technique) is extracted from the resultant Pareto front in the 30 independent executions performed, and a Student's t-test with 5% confidence level is performed over the difference of those values, determining whether the difference is statistically significant or not. If the difference is statistically significant, the best technique is indicated, including the '-' symbol in any other case.

The comparison of the results for the introduced technique and the ones contained in table 6.9 is presented in figures 6.14-6.16. Each of these figures presents the graphical comparison on a different curve from the dataset. The cross marker indicates the mean result from the

Table 6.7: Freeman chain code representation of the figures in the dataset

Chromosome
00665 65560 01010 76555 45555 55555 43112 12233 45432 01101 11112 11212
Leaf
66656 65500 01005 66565 50011 06656 56555 55666 76666 66666 42222 22222 22322 44343 33333 23070 00033 23230 70000 33232 22677 72221 27666 11111
Semicircle
00007 00777 77766 76666 66665 76766 56454 43436 66656 55454 44434 33232 22254 54434 23221 21322 22222 21221 11111 0010000

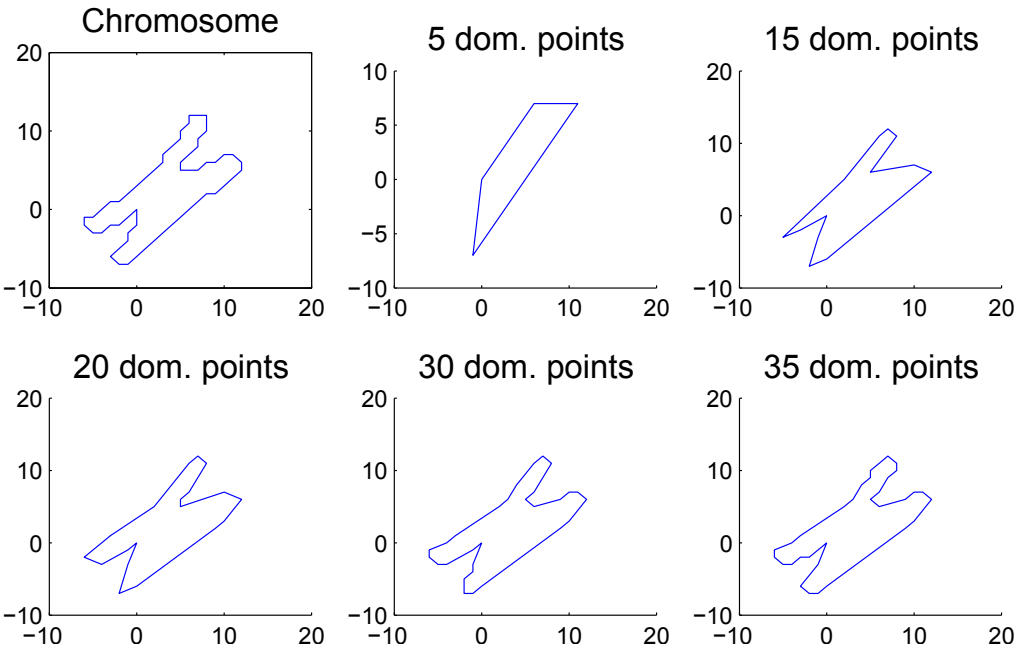


Figure 6.11: Chromosome curve

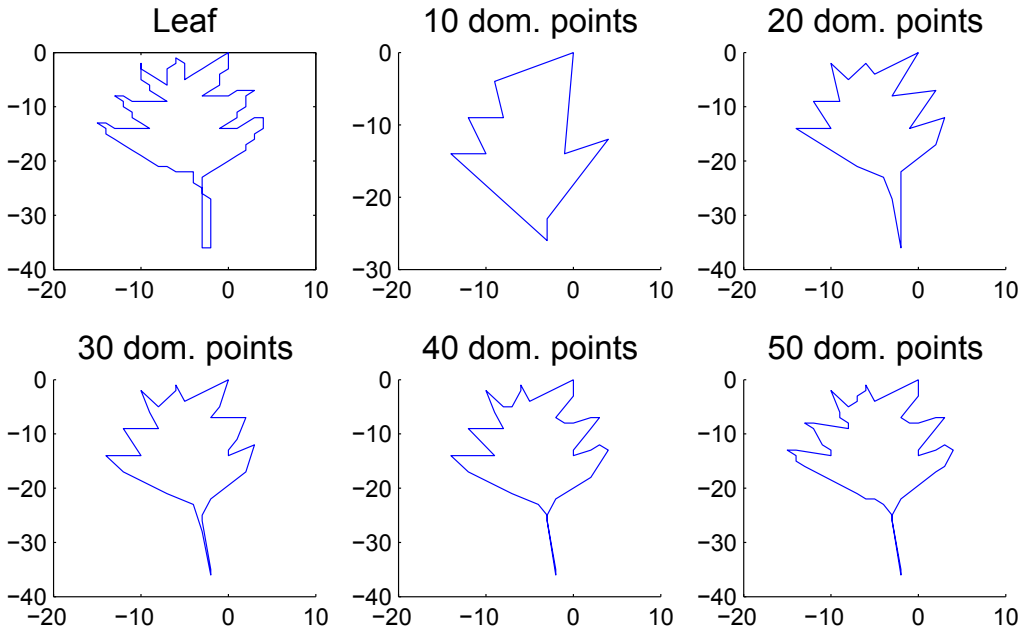


Figure 6.12: Leaf curve

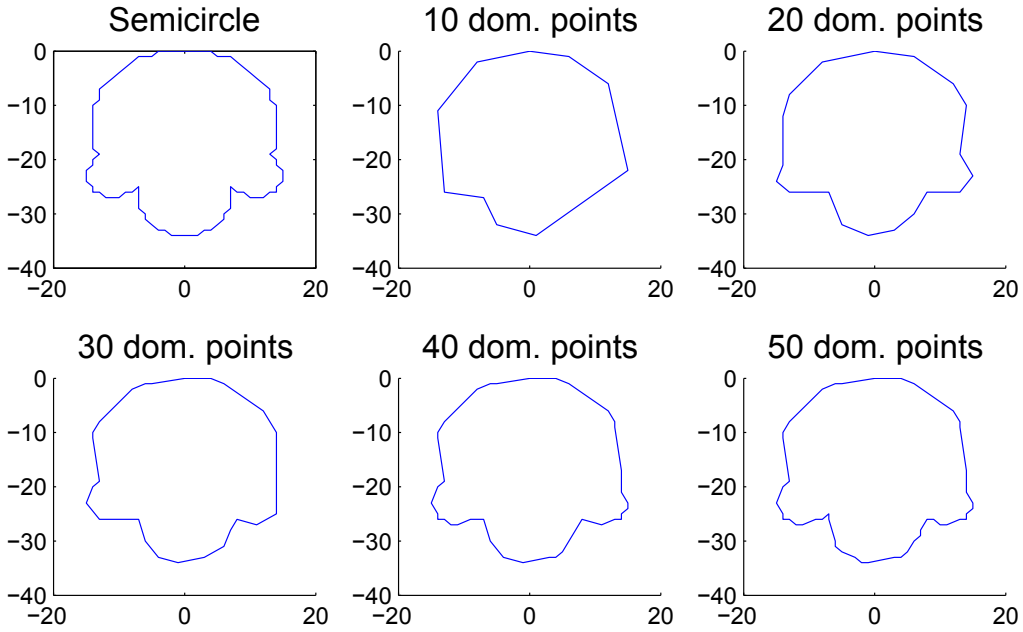


Figure 6.13: Semicircle curve

Table 6.8: Pareto Front dominant points / integral squared error results for the dataset

Dom points	Chromosome		Semicircle		Leaf	
	Mean	Std	Mean	Std	Mean	Std
1	7037 33	1329 14	88648 00	0 00	56705 63	10709 98
2	502 67	94 94	11200 00	0 00	3682 00	695 42
3	343 57	64 89	2436 10	38 57	437 78	82 68
4	133 13	25 14	1093 39	19 67	342 40	64 73
5	96 75	18 56	559 70	30 86	288 78	55 04
6	25 18	4 76	142 49	1 43	217 89	41 57
7	18 84	3 56	116 43	1 31	183 14	34 71
8	12 99	2 45	91 05	1 34	154 78	29 69
9	11 68	2 21	74 07	2 29	134 69	25 54
10	7 80	1 47	60 51	1 64	107 29	20 32
11	6 83	1 29	39 76	3 90	89 89	17 11
12	5 62	1 06	30 71	2 33	63 60	12 79
13	4 65	0 88	23 58	1 48	47 76	9 33
14	4 03	0 76	17 90	0 79	42 45	8 06
15	3 67	0 69	14 60	0 57	31 26	6 14
16	3 34	0 63	13 54	0 11	26 10	4 99
17	3 03	0 57	12 30	0 22	21 17	4 05
18	2 74	0 52	11 23	0 06	16 14	3 07
19	2 47	0 47	10 10	0 10	14 64	2 77
20	2 28	0 43	9 06	0 07	13 05	2 47
21	2 08	0 40	8 10	0 07	11 59	2 20
22	1 89	0 36	7 19	0 09	10 35	1 96
23	1 70	0 32	6 34	0 15	9 19	1 74
24	1 50	0 29	5 57	0 22	8 39	1 59
25	1 31	0 25	4 97	0 26	7 75	1 47
26	1 13	0 22	4 51	0 15	7 13	1 35
27	1 00	0 19	4 15	0 11	6 51	1 23
28	0 88	0 17	3 80	0 11	6 00	1 13
29	0 77	0 15	3 45	0 11	5 54	1 05
30	0 65	0 12	3 10	0 11	5 08	0 96
31	0 54	0 10	2 75	0 11	4 69	0 89
32	0 46	0 09	2 51	0 09	4 33	0 82
33	0 34	0 06	2 32	0 08	3 98	0 75
34	0 30	0 06	2 15	0 08	3 63	0 69
35	0 15	0 03	1 98	0 07	3 37	0 64
36	0 15	0 03	1 79	0 08	3 13	0 59
37	0 00	0 00	1 63	0 07	2 87	0 54
38			1 47	0 06	2 63	0 50
39			1 28	0 08	2 37	0 45
40			1 17	0 03	2 14	0 41
41			1 08	0 00	1 95	0 37
42			0 93	0 02	1 75	0 33
43			0 86	0 00	1 56	0 30
44			0 77	0 00	1 38	0 26
45			0 62	0 00	1 21	0 23
46			0 62	0 00	1 05	0 20
47			0 46	0 00	0 90	0 17
48			0 46	0 00	0 77	0 15
49			0 31	0 00	0 67	0 13
50			0 31	0 00	0 53	0 10
51			0 15	0 00	0 44	0 08
52			0 15	0 00	0 30	0 06
53			0 00	0 00	0 28	0 05
54					0 15	0 03
55					-	-
56					0 00	0 00

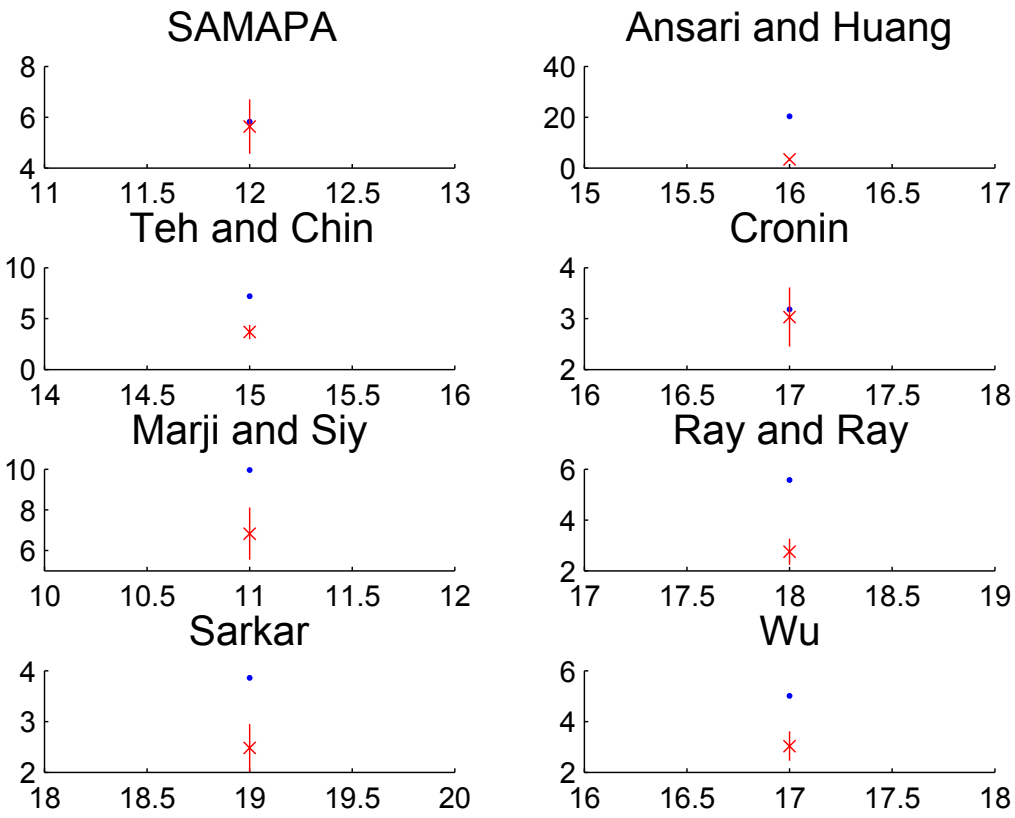


Figure 6.14: Chromosome results comparison

MOEA technique, crossed by a vertical line delimiting the standard deviation boundaries, whereas the solid dot is the result from the concrete technique which it is compared to. The chosen element from the obtained Pareto Front is the one with the same number of dominant points as the solution produced by the concrete technique it is compared to (in a procedure similar to the one followed for the statistical comparison).

The statistical comparison shown in table 6.10 determines that the MOEA technique is significantly better than the other alternatives in 21 out of 24 test cases, being significantly worse only in one case (Cronin's result for the semicircle curve). Also, the differences between its results and the alternatives are very significant, which can be observed in the different graphical comparisons presented in the figures and the low p-values contained in the tables. The dataset is rather scarce, but without standard implementations of the techniques or a framework to properly test them with novel data, the comparison has resorted to the results in their reference papers, which only included these figures. The good performance results of the evolutionary technique against a set of techniques specialized for this particular domain are, in any case, remarkable.

The evolutionary technique presented in (Yin, 1999) did not provide results for the leaf curve, so the comparisons will be focused on the remaining two figures of the dataset. Also, as explained in section 6.2.3, several solutions with different numbers of dominant points (building artificially a set of solutions similar to a Pareto Front) are presented for each of the curves in its dataset. Tables 6.11 and 6.12 present these results for the chromosome

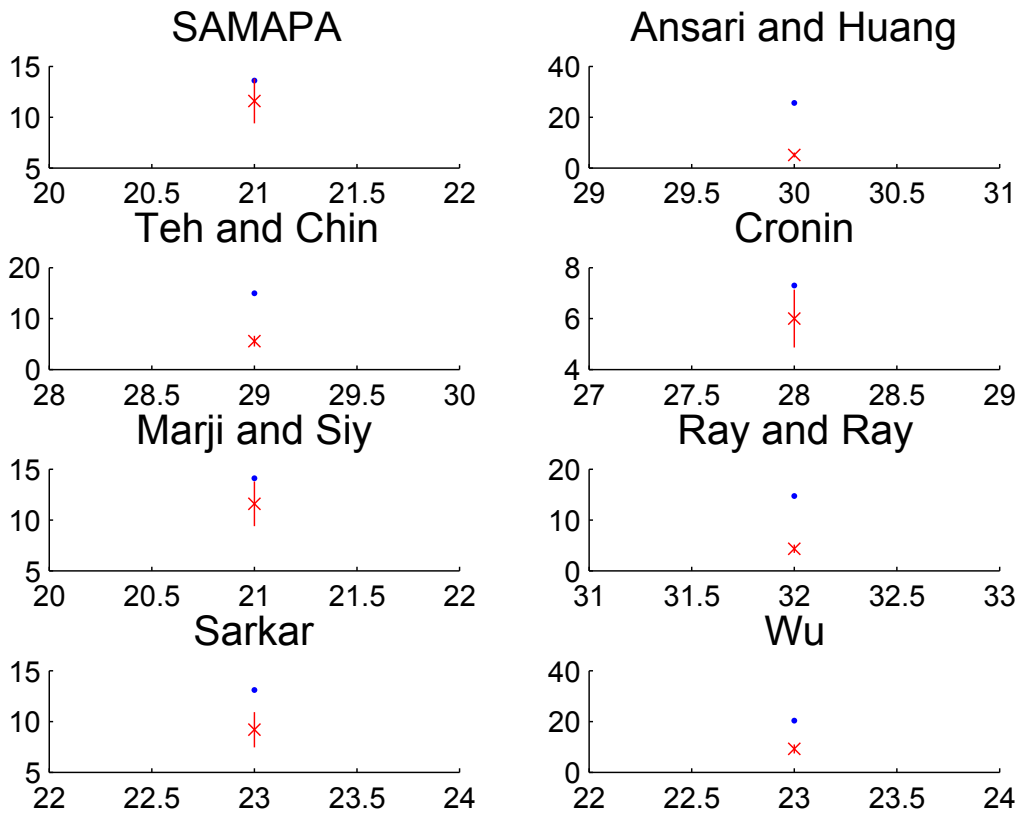


Figure 6.15: Leaf results comparison

Table 6.9: Comparable techniques results for the dataset

Technique	Chromosome		Semicircle		Leaf	
	Dom. Points	ISE	Dom. Points	ISE	Dom. Points	ISE
SAMAPA	12	5.82	19	12.90	21	13.60
Ansari and Huang	16	20.30	28	17.80	30	25.60
Teh and Chin	15	7.20	22	20.60	29	14.96
Cronin	17	3.18	30	2.91	28	7.30
Marji and Siy	11	9.96	18	24.20	21	14.10
Ray and Ray	18	5.57	29	11.80	32	14.70
Sarkar	19	3.86	19	17.40	23	13.10
Wu	17	5.01	27	9.01	23	20.34

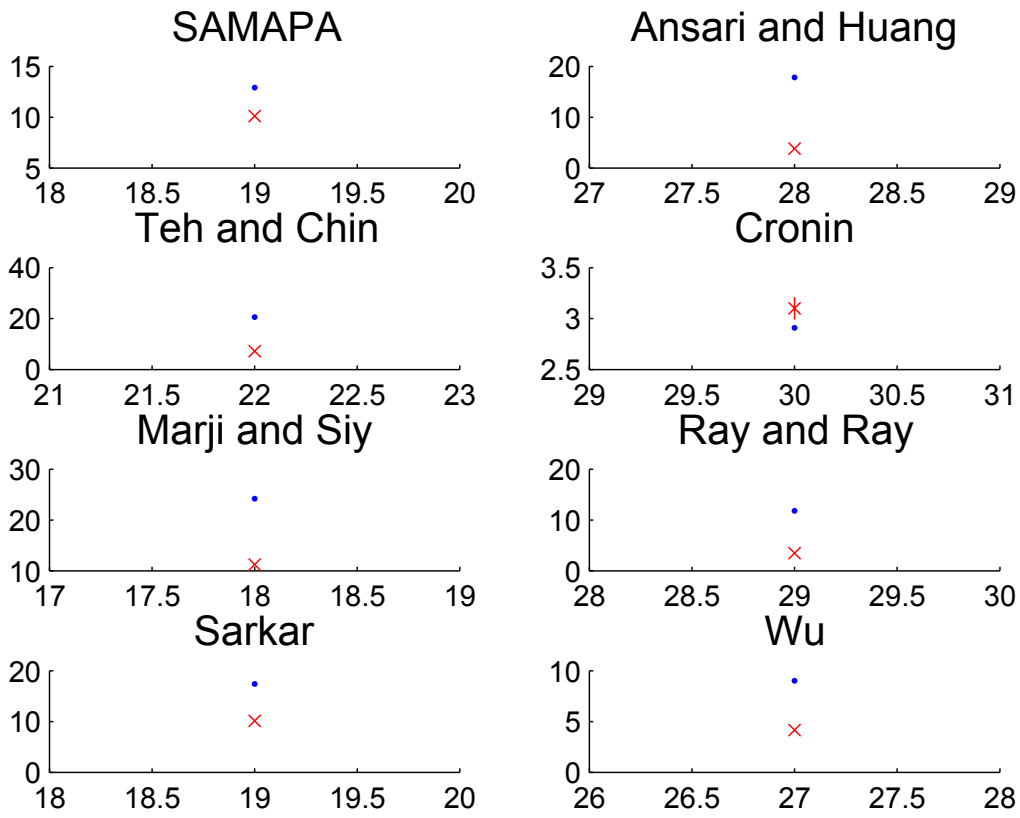


Figure 6.16: Semicircle results comparison

Table 6.10: Statistical result comparison

Technique	Chromosome		Semicircle		Leaf	
	p-value	stat. best	p-value	stat. best	p-value	stat. best
SAMAPA	3.21E-01	-	1.67E-43	MOEA	2.52E-05	MOEA
Ansari and Huang	3.15E-43	MOEA	4.55E-63	MOEA	2.27E-40	MOEA
Teh and Chin	1.80E-22	MOEA	3.68E-64	MOEA	1.57E-29	MOEA
Cronin	1.54E-01	-	2.09E-10	Cronin	7.23E-07	MOEA
Marji and Siy	6.93E-14	MOEA	3.43E-70	MOEA	8.06E-07	MOEA
Ray and Ray	2.47E-23	MOEA	2.20E-56	MOEA	8.18E-34	MOEA
Sarkar	4.36E-16	MOEA	1.47E-55	MOEA	4.75E-13	MOEA
Wu	6.62E-18	MOEA	1.67E-49	MOEA	2.54E-25	MOEA

Table 6.11: Yin's results and statistical comparison for the Chromosome curve

Dom. Points	ISE	p-value	stat. best
8	17.41	8.67E-11	MOEA
9	13.82	1.02E-05	MOEA
12	7.99	6.06E-13	MOEA
14	5.47	2.82E-11	MOEA
15	5.22	6.00E-13	MOEA
17	4.58	4.17E-15	MOEA
18	4.17	2.93E-15	MOEA

Table 6.12: Yin's results and statistical comparison for the Semicircle curve

Dom. Points	ISE	p-value	stat. best
10	52.95	2.73E-21	Yin
12	42.85	8.77E-23	MOEA
14	29.93	4.42E-36	MOEA
17	17.41	3.80E-41	MOEA
18	14.80	6.21E-54	MOEA
19	14.94	2.19E-50	MOEA
22	12.91	1.99E-53	MOEA
27	7.04	5.70E-43	MOEA
30	6.61	1.17E-45	MOEA

and semicircle curves respectively. In those tables, an statistical significance test is also presented, treating each of the solutions provided by Yin's algorithm individually and with the same parameters used for the comparison with the previous techniques.

Additionally, a statistical comparison from a multi-objective perspective has been carried out. This has been performed with the extraction of the Pareto fronts contained in Yin's solutions (removing one dominated solution) and the computation of the hypervolume values from those fronts. These hypervolume computations required the choice of the corresponding *nadir* points (the worst possible solution points). These points have been adapted to the portion of the Pareto front covered by the solutions. For the chromosome curve, their values are 19 dominant points with 19.15 ISE and for the semicircle curve 31 dominant points with 66 ISE. To obtain these values, 1 was added to the maximum value of dominant points in the fronts, and an additional 10% to the maximum ISE value. To obtain the hypervolume values from the MOEA solutions, one individual is extracted for each one provided by the artificial Yin Pareto front it is being compared to, building a distinct Pareto front for each execution. Finally, a t-test is run over the hypervolume results to test their statistical significance. It must be noted that the different MOEA runs provided solutions for dominant point values not contained in Yin's solutions, which were not included to make the comparison fairer.

The comparison to Yin's results is shown in figures 6.17 and 6.18. These figures present the whole section of our Pareto Front according to the highest and lowest number of dominant points presented in Yin's results. This section of the Pareto Front contains more points than the ones included for the statistical multi-objective comparison, according to the procedure

Table 6.13: Muli-objective hypervolume comparison from the reduced Pareto front and Yin's algorithm

Curve	MOEA mean	MOEA std	Yin	p-value	stat. best
Chromosome	0.3690	0.0397	0.3055	1.19E-09	MOEA
Semicircle	0.5089	0.0036	0.4573	3.23E-35	MOEA

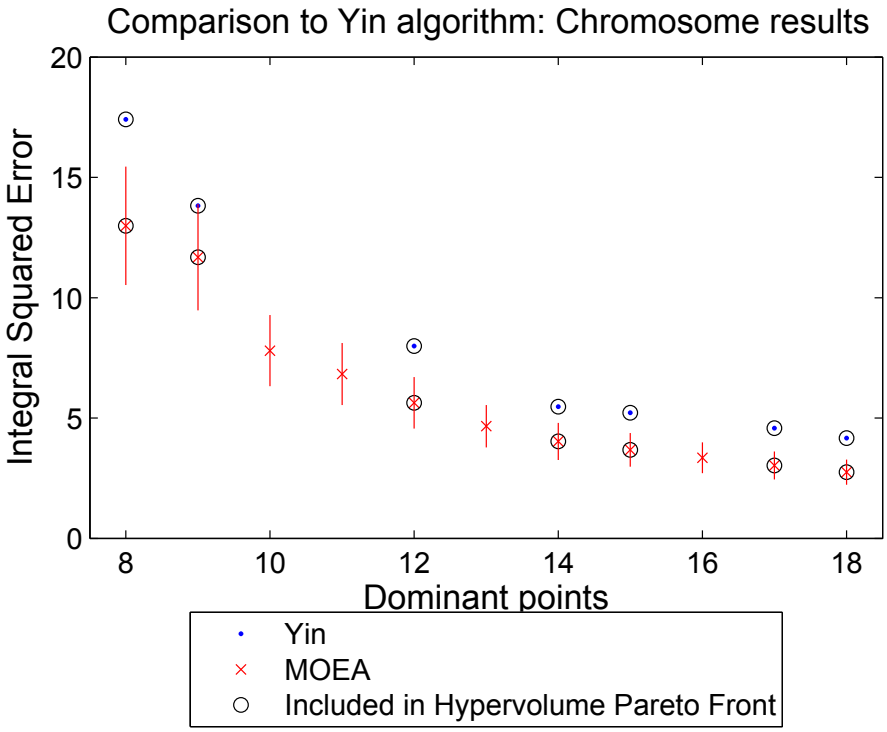


Figure 6.17: Yin's chromosome results comparison

previously explained, where only those points with a number of dominant points contained in non dominated Yin's solutions were included. According to that comparison, the points included in the quality indicators for those Pareto Fronts have been highlighted with a circle marker. The individual comparison to Yin's results shows significantly better results for the MOEA technique in 15 out of 16 cases. This statistical difference is corroborated with the hypervolume results, where the MOEA technique is significantly better in both curves.

The overall results show that the MOEA solution to the segmentation is issue is extremely competitive with the available works in the literature in terms of quality of the obtained solutions in the Pareto Front.

6.6 Conclusions

This work has been focused on the segmentation issue by means of Piecewise Linear Representation, which is present in the polygonal approximation domain, highlighting its unresolved issues. One of those issues is the multi-objective nature of segmentation processes, where

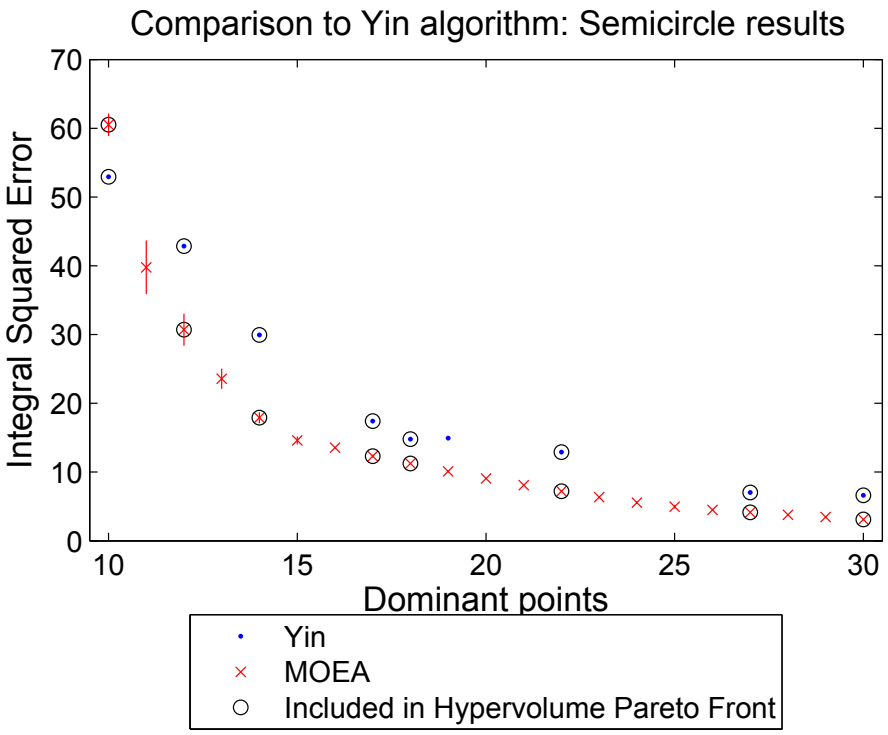


Figure 6.18: Yin's semicircle results comparison

several objective functions have to be optimized jointly. This fact has not received the proper attention in terms of algorithm development (only for certain comparison purposes). Even so, any technique available has to deal with this multi-objective nature of the problem, even if this nature is not explicitly declared. Four representative algorithms have been detailed, covering their implicit treatment of that multi-objective nature, based on a-priori approaches. This discussion has lead to the explicit formulation of segmentation as a proper multi-objective problem and its resolution by means of an a-posteriori approach using a multi-objective evolutionary algorithm. For the results presentation, the chosen algorithm is SPEA2, along with default transformation operator values. The segmentation domain characteristics, along with the representation used, allow the introduction of a specialized initialization operator which, in order to improve the algorithm performance, obtains an initial population with a better coverage of the search space. The population size and number of generation values are chosen according to Wilcoxon test results over a set of possible configurations with increasing values.

The final objective of the multi-objective evolutionary approach is obtaining the whole Pareto front of possible segmentation results for a given problem. Parametric techniques can obtain artificial Pareto fronts with several different runs configured with different input parameters, being each of these solutions independent. This is computationally inefficient and can lead to additional optimization problems (such as the determination of the proper error approximation value in order to obtain a certain number of segments in the solution). These problems are inherently solved with the use of the MOEA approach presented in this work. Also, the different solutions in the Pareto front of a segmentation problem share valuable information in the form of dominant point position, leading to faster and better

solutions when compared to obtaining individual elements from that Pareto front.

The results obtained in the Pareto front with the chosen technique in the polygonal approximation dataset used are extremely competitive with the available works in the literature, having obtained statistically significant improvements in 36 out of the 40 individual results, and also in the two curves compared under a multi-objective perspective by means of the hypervolume quality indicator, showing that treating the multi-objective nature of the problem explicitly allows the algorithm to obtain better solutions. It is important to highlight that this technique is able to cope with the requirements presented, allowing the final user to regain its role as the decision maker of the problem and to change which solutions fit its requirements at different moments (provided by obtaining the whole Pareto Front in a single execution).

The proposal, even though it has been able to prove the advantages of the multi-objective evolutionary approach, does not completely fulfill the thesis requirements. The stopping criterion has been established a-posteriori, according to the quality results of the obtained solutions, which have lead to the configuration parameters. These configuration parameters, according to this setting procedure, are specific to the problem instances faced (or different ones of similar difficulty), but have not been tested for additional problem instances. Even among the tested dataset, the number of generations (lacking a dynamic stopping criterion, such as the ones presented in chapters 4 and 5) is not the optimal one for all the problem instances (there were no differences in the final hypervolume results for the easiest curve, the chromosome, between 1000 and 2000 generations). Thus, there are required additional considerations and adaptations to be performed over this general MOEA approach, which will be faced in chapter 7.

7

An efficient approach to multiobjetive evolutionary polygonal approximation

> *Well, Diotallevi and I are planning a reform in higher education A School of Comparative Irrelevance, where useless or impossible courses are given The school's aim is to turn out scholars capable of endlessly increasing the number of unnecessary subjects*
>
> Umberto Eco, *Foucault's Pendulum*, 1988

This chapter will deal with the design features of applying a multiobjective evolutionary approach to the polygonal approximation domain. While chapter 6 presented the underlying basis for the technique and tested its capabilities versus a significant set of available techniques from the domain, the computational complexity was not dealt with. This chapter will face this issue from a number of different points of view: from the initialization process to the application of the designed stopping criterion, dealing with the complexity of the fitness function and the required modifications performed to the transformation operators in order to simplify this complexity. The main reference works for this chapter are (Guerrero et al., 2014b, 2012a, 2013a,b)

7.1 Local fitness computation and fitness-aware transformation operators

As introduced in chapter 6, two different problems can be presented regarding polygonal approximation: $Min - \#$ and $Min - \epsilon$, which differ in the objective of the segmentation process: minimizing the number of segments in order to obtain a representation error lower than a certain threshold or, on the other hand, minimizing the representation error for a given number of segments. Evolutionary approaches to this domain have been focused on solving the $Min - \#$ problem, which forces the use of specific transformation operators (crossover and mutation) in order to not modify the number of dominant points in a given solution. For a given curve to be segmented, its codification α is a string of $0's$ and $1's$, determining whether each of the points is considered dominant. The fitness presented was- based on two different representation errors widely extended and used by different techniques, $E_\infty(\alpha)$ and

$E_2(\alpha)$ (usually referred to as maximum error and integral square error, respectively). These errors are represented in equations 7.1 and 7.2

$$maxError = max_{i=1}^{n} e_i \tag{7.1}$$

$$ISE = \sum_{i=1}^{n} e_i \tag{7.2}$$

Genetic algorithms (as seen in section 2.4) are based on a series of basic steps performed every generation over a certain population of individuals. An initial population is generated and the evolutionary cycle starts with the application of a series of operators: selection of individuals who will receive the application of transformation operators, application of these operators (usually crossover and mutation), fitness update of the newly generated individuals and finally the selection of next generation's population, being this procedure followed until a certain stopping criterion is triggered. Under this general approach, domain knowledge was introduced in the fitness computation steps and the remaining actions could be faced as exchangeable black boxes, allowing independent research in each of these boxes.

However, the importance of introducing specific domain information at different steps has to be considered for practical application of evolutionary algorithms. In fact, for current domain, section 6.4.2 already tried to introduce some domain information in order to provide more efficient initializations based on specific heuristic local search procedures, even if these procedures did not lead to improved results in the final front solutions.

The motivation for this section is to understand the effect of transformation operators over the produced offspring, relating their fitness computation to that of their parents. A simple example regarding this relationship is shown in figure 7.1. This figure shows the effect of a mutation which actually changes the value of a single point from the curve and the local effect which it produces over the fitness computation. Both genotype and phenotype have been included to clarify the followed representation. As shown in the example, the mutation (which creates a new dominant point) does only imply the new calculation of two segments. The remaining parameters and segment errors remain unchanged from the previous chromosome. However the fitness computation would recalculate all the segmentation parameters and individual errors in order to calculate the representation error, since the genotype has no information regarding the partial fitness computations of the original parent.

Since the fitness is based on the aggregate value (summation) of the errors of the different individual segments, storing that information would allow the introduction of fitness-aware operators able to recalculate only the minimum amount of required information. This approach implies the storage of the errors for the individual segments, in order to perform only partial fitness computations. The chromosome for each solution instance will thus have an additional array of floating point values storing the approximation error value of the given segment (if the point is dominant) or zero (in any other case). Following this representation, the mutation procedure presented in figure 7.1 produces only local changes, as shown in the genotypes represented in figure 7.2.

This partial fitness storage on the chromosome will introduce a fitness function which would simply have to sum the different values in the complementary chromosome structure in order to obtain the final fitness value. Since the changes produced by the transformation operators are local, the whole fitness value can be recomputed locally (and not only the values in

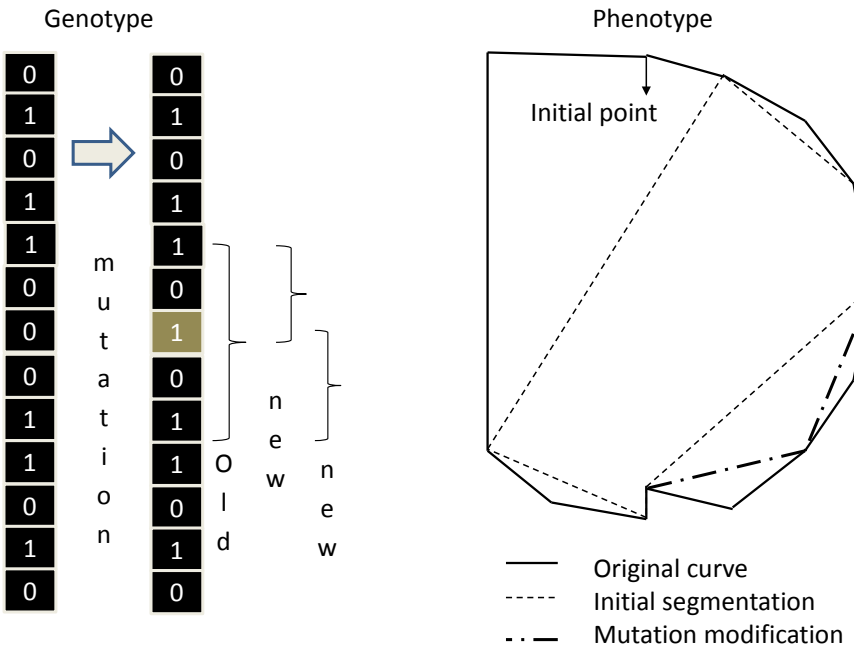


Figure 7.1: Representation of the local changes produced by a mutation operator

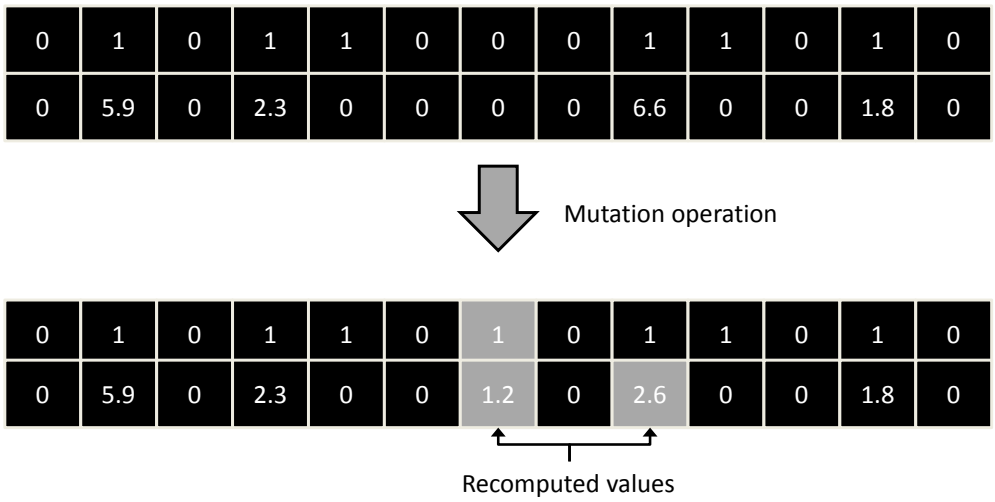


Figure 7.2: Partial fitness values recomputation derived from the local changes introduced by a mutation operator

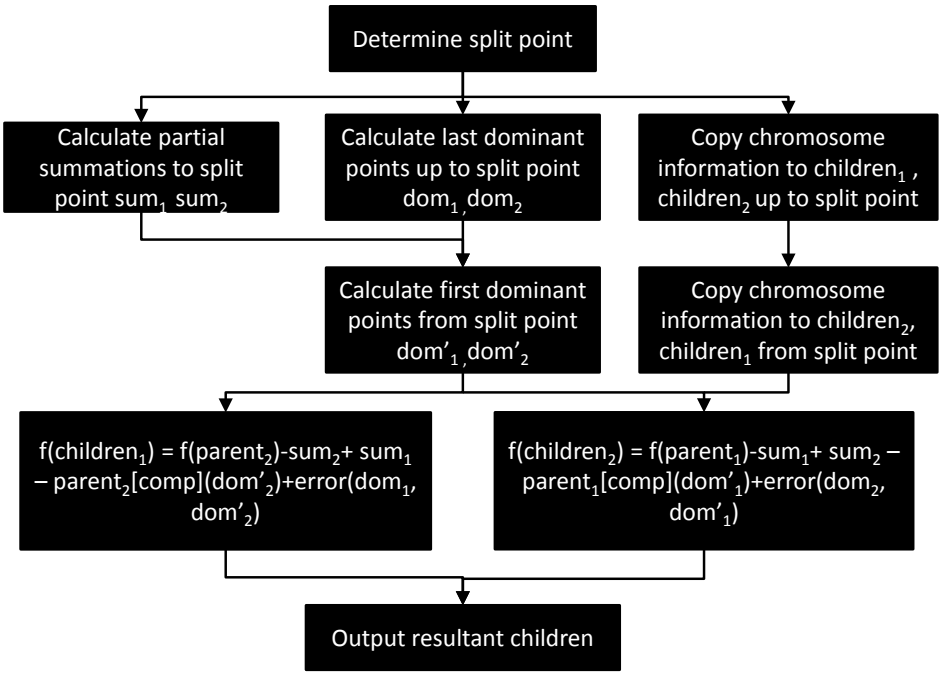


Figure 7.3: Pseudoalgorithm for the 1-point crossover fitness-aware operator

the complementary chromosome structure). Transformation operators which perform partial fitness recomputations to calculate the objective values are defined as fitness-aware operators. Following the operator choices presented in chapter 6 (the summary of the proposal was presented in table 6.3) figures 7.3 and 7.4 show 1-point crossover and bitflip mutation for the proposed representation. In these operator figures, the straightforward computation of the number of dominant points has not been included to improve the readability.

These fitness-aware operators also apply a recomputation of the whole fitness value, which is stored in the chromosome. This is performed in order to further simplify the fitness computation process, bearing in mind that the length of the problems can be quite large (such as those presented in (Kolesnikov, 2012)), and even the sum of the partial fitnesses can take a sizable amount of time. This fitness value can be computed more efficiently at the application time of the transformation operators, since they can handle valuable information regarding the parents partial fitness values which become unavailable once the transformation operations have ended. Figures 7.5 and 7.6 show an example of how these fitness values are calculated.

Profiling the resultant algorithm, the mutation procedure started to take up a much higher percentage computational cost than crossover, particularly when applied to problem instances with a large number of data. This computational cost was increased due to the required pseudorandom number generation for each gene position in order to determine whether it needed to be mutated or not.

The proposed alternative mutation procedure is based on two steps: the first time the mutation procedure is used, a set of random numbers equal to twice the chromosome size

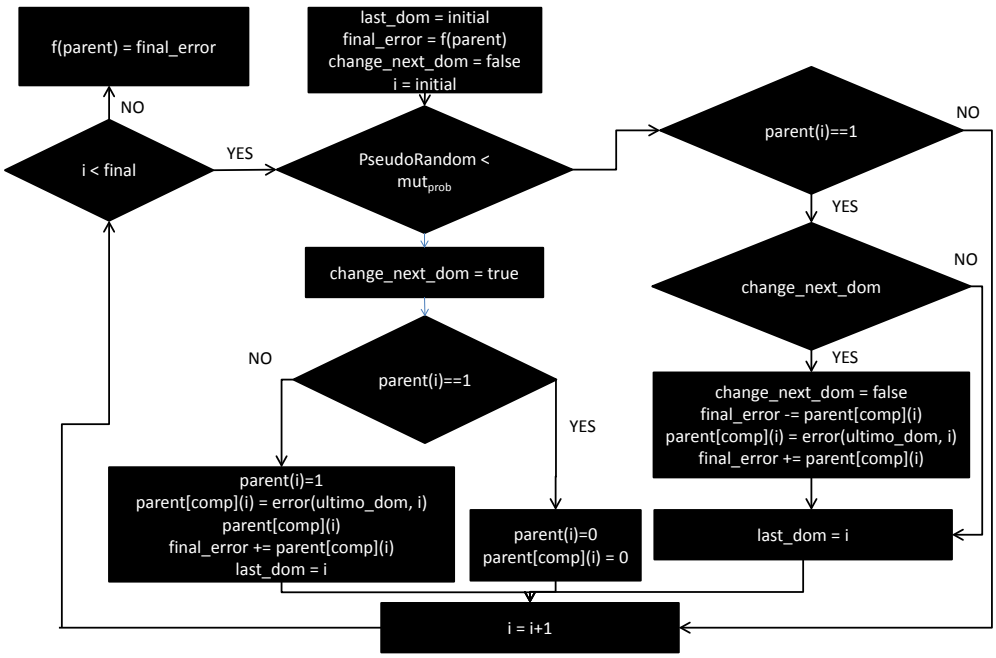


Figure 7.4: Pseudoalgorithm for the mutation fitness-aware operator

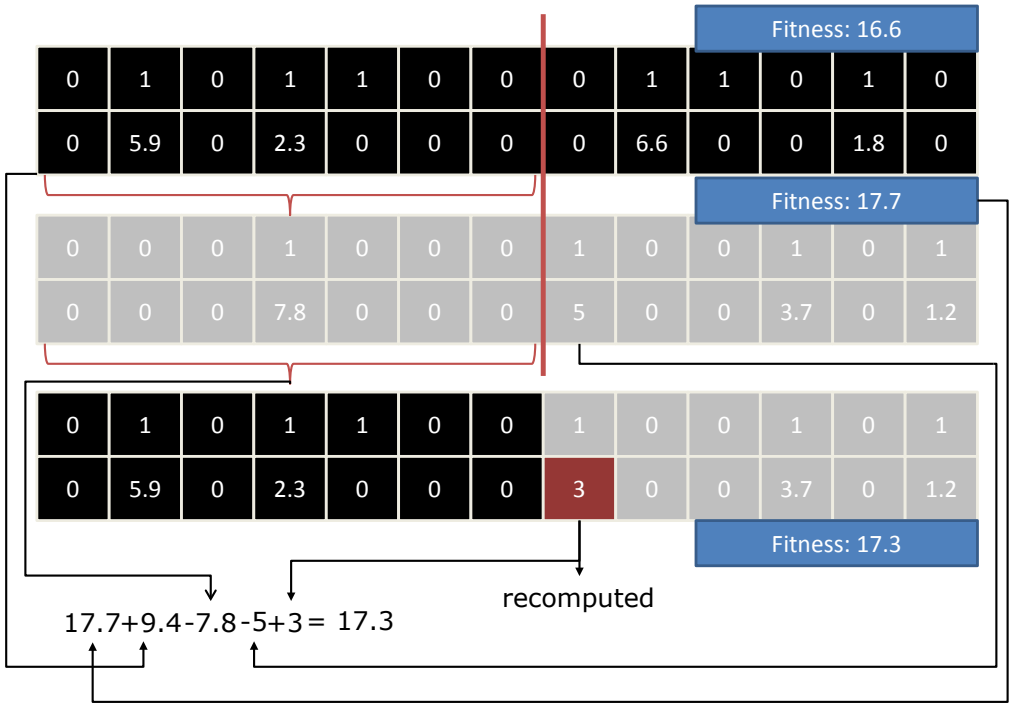


Figure 7.5: Child individual fitness computation by crossover fitness-aware operator

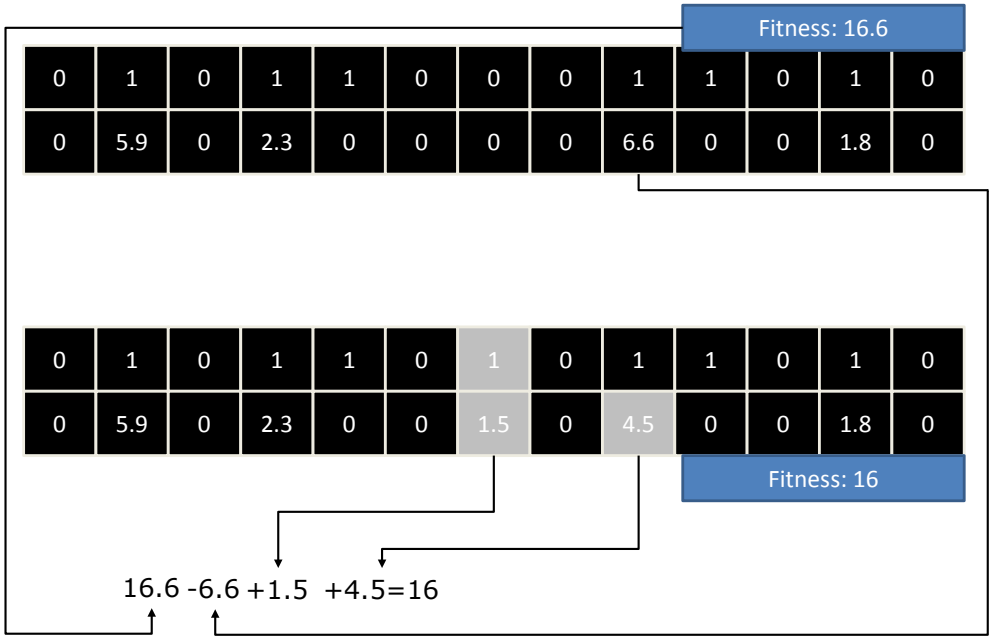


Figure 7.6: Child individual fitness computation by mutation fitness-aware operator

is calculated and stored. Then, every time the mutation procedure is used, a single random number is calculated to determine the starting position over the pre-calculated vector, and the pseudorandom values required are sequentially obtained from the required positions starting at that random value. A study regarding the influence of random number generators in evolutionary algorithms may be found in (Cárdenas-Montes et al., 2012).

For a problem such as the leaf curve (with 120 points) and the parameters set in the proposed configuration shown in table 6.3, with 500 individuals, 2000 generations and 1/n mutation probability, the original value of psedorandom floating point number of values to be generated was, roughly, $12 * 10^7$. With the proposed approach, this number is reduced to 240 pseudorandom floating points (the initial values of twice the chromosome length) plus 10^6 pseudorandom integer values to determine the initial value each time the mutation operator is run (which implies a difference of two orders of magnitude).

7.2 An alternative archiving technique

7.2.1 Overview over archiving techniques

In (Laumanns et al., 2002), Laumanns et al proved that many MOEAs based on standard Pareto-based selection schemes could suffer deterioration, not guaranteeing convergence. Deterioration occurred when elements of a solution set at a given time were dominated by a solution set which the algorithm maintained some time before. Based on these observations, they presented new archiving strategies based on the ϵ-dominance concept, attempting to provide both convergence and good distribution properties.

However, in (Knowles & Corne, 2003), some of the issues with this approach were

highlighted, mainly the choice of the ϵ initial parameter. This parameter could be chosen by either a preset value or by an adaptative procedure. In the former case, the number of points in the archive is bounded by a function of the (possibly unknown) objective space ranges. In the latter case, ϵ may become arbitrarily large, providing a poor final archived set compared to the sequence of points presented to the archiving algorithm.

SPEA algorithm family, both SPEA (Zitzler & Thiele, 1999) and SPEA2 (Zitzler et al., 2001), rely on the concept of strength for their archiving strategy: originally proportional to the number of solutions which an individual dominated (in the SPEA algorithm), it was improved in the SPEA2 algorithm by also including the number of solutions which dominate it. This led to the environmental selection update mechanism for the archive (an analysis of the algorithm was covered in section 6.4.3).

The original archive update mechanism was based on a clustering technique. This mechanism tended to lose boundary solutions when the archive size was too small for the required number of non-dominated solutions. The truncation technique presented in SPEA2 is an iterative process which eliminates at each stage the individual with the minimum distance to another individual (considering the following distances to the second, third... closest individuals in case of ties). This process continues until the maximum number of individuals according to the archive size have been introduced.

The archive size in SPEA2 is fixed. If the number of non-dominated individuals is not sufficient to fill it, dominated ones are inserted. Also, the environmental selection mechanism dominates the complexity of the whole algorithm, with a worst case complexity of $O(M^3)$, where M is the population size plus the archive size. On average, that complexity is reduced to $O(M^2 logM)$.

7.2.2 An alternative archiving procedure

Section 7.1 has faced the complexity of the fitness function for the algorithm, also profiling and dealing with issues related to the fitness-aware operators introduced. The overall computational cost distribution is heavily altered by these changes. If we compare the cost of the archiving technique vs the whole remaining operations for every evolutionary cycle, the result is presented in figure 7.7.

These results have been obtained using the JMetal (Durillo & Nebro, 2011) environment with the general configuration established in table 6.3 using the enhanced fitness computation and fitness aware operators described in section 7.1. As figure 7.7 clearly shows, the archiving procedure is not only dominating the algorithm running time, but also the enhanced fitness computation (archiving implies more the 99% of the whole running time, including fitness computation). This huge effort to guarantee a well distributed Pareto front seems unacceptable. Therefore, an alternative archiving procedure needs to be introduced.

The polygonal approximation process has a set of very specific characteristics, mainly its bi-objective nature with a very high degree of conflict between them and the fact that one of these objectives is discrete. Figure 7.8 shows the result of an initialization process prior to the application of Pareto dominance selection to highlight these characteristics.

Some of the issues related to the costly archiving results exhibited by SPEA2 in this problem, as shown in figure 7.7 are related to the multi-objective proposal: the algorithm must be able to store, ideally, one individual per each compression level. This implies that the archive size can get to be really large (and the computational complexity of the environmental

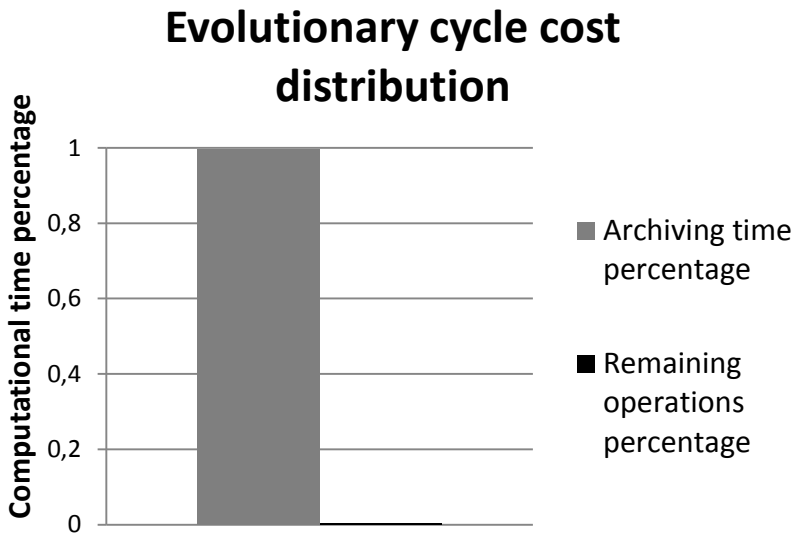


Figure 7.7: Computational cost distribution between archiving technique and the remaining procedures of an evolutionary cycle

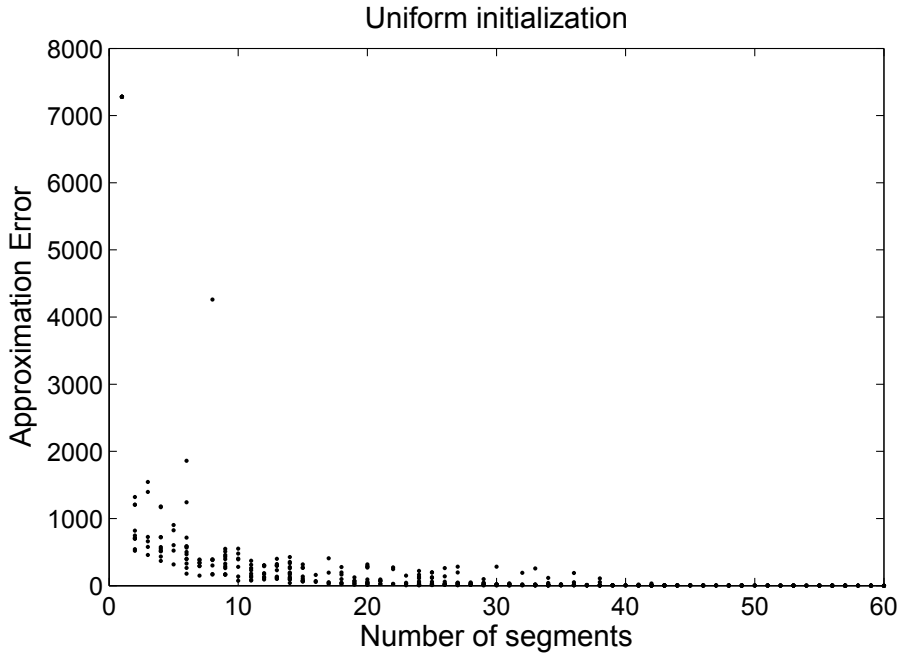


Figure 7.8: Initialization example showing dominated and non-dominated individuals

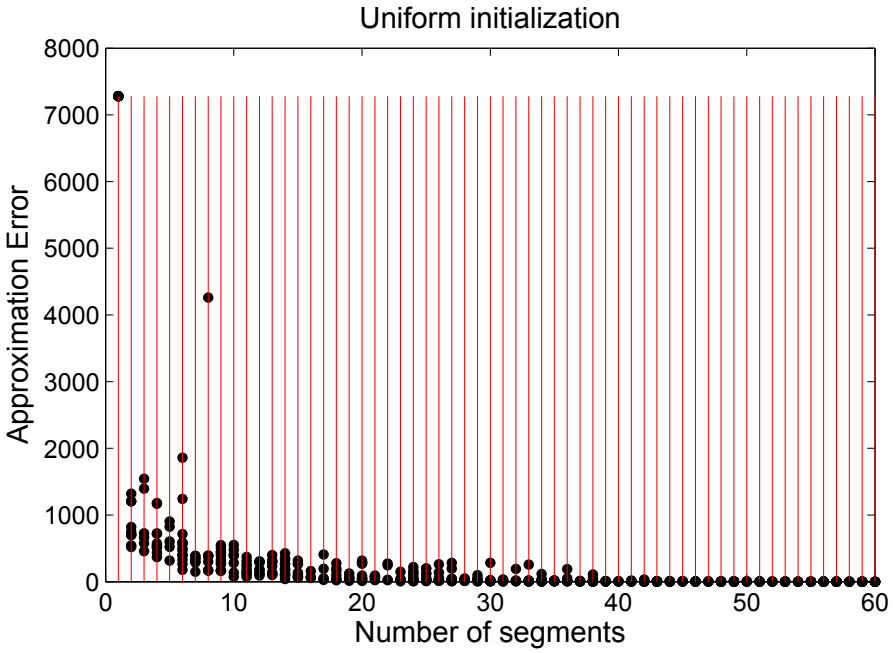


Figure 7.9: ϵ glitches over the population initialization

selection depends heavily on that archive size value). On the other hand, this provides a boundary for this size (a value which can get to be difficult to establish a-priori).

The concept of $\epsilon - dominance$ was introduced in definition 2.6.1 along with its use in the epsilon indicator (described in section 5.2.1). The idea presented in (Laumanns et al., 2002) was, according to $\epsilon - dominance$, to draw ϵ-boxes such that at most one element is contained in each box. From the characteristics of the problem presented, the idea for the alternative archiving presented is to use a technique similar to these ϵ-boxes, considering a box for each of the possible individuals according to the number of segments objective. Figure 7.9 represents these boxes over the results in figure 7.8.

As shown in figure 7.9, these boxes are infinitely thin in one objective (the discrete objective representing the number of segments, where they only cover one value) and infinitely long in the other (the objective representing the representation error). These particular instances of ϵ-boxes are similar to the glitches from signal processing theory, and so they have been named according to this resemblance. Pareto-dominance is only checked within the ϵ-glitch which an individual belongs to, not among different glitches. This implies that the complexity of this process is now constant, and the complexity of the whole archiving mechanism is reduced to $O(n)$, where n is the population size. Figure 7.10 shows the result of the ϵ-glitches archiving procedure over the population presented in figure 7.8.

It must be noted that one of the issues related to the environmental selection process was the fact that its complexity order included the archive size, This issue has been overcome, a specially important achievement for this problem, since the required number of individuals in the archive can be very large for some problem instances, as has been repeatedly noted.

The traditional evolutionary cycle where one full generation is produced at each step is no longer required, since every individual is compared to the correspondent one in its glitch already stored in the archive. With this approach, the evolutionary cycle implies parent

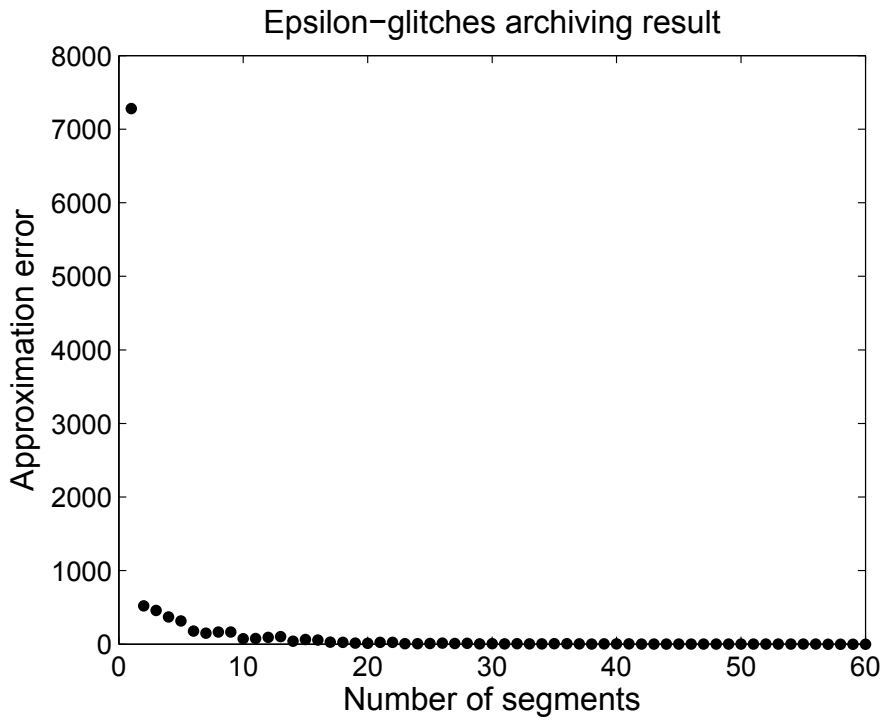


Figure 7.10: Result of the ϵ glitches archiving technique over the population initialization

selection, children obtaining through the transformation operators and the invocation of the archiving procedure individually.

Even though implementation issues are not the focus of this work, the appropriate management of the data structure for the archive is important for the computational cost reduction. Since the archive size is fixed (with an exception detailed in the following paragraph), a fixed size, constant time access data structure is suggested (such as a traditional array). For the initialization of this array, we suggest the generation of an initial random individual for one of the archive boundaries (either of them) and the application of directed mutations to obtain an individual for each of the ϵ-glitches. These directed mutations (and, thus, the archive initialization process) imply a very low computational cost using the fitness computations detailed in section 7.1.

One final improvement is introduced in the archiving procedure. For the MOEA proposal to polygonal segmentation, the initial archive size, as explained in chapter 6 is set to the number of points in the curve. However, very commonly through the evolutionary process, new individuals with a perfect segmentation (zero error) are found, requiring a lower number of segments than that initial boundary. This effect can be seen in the results presented in table 6.8. The initial archive size for the leaf curve was 120, but an individual with only 56 dominant points can achieve a perfect segmentation.

With the default SPEA2 archiving mechanism, this increased archive size adds complexity to the algorithm, due to the relationship between the archive size and the procedure computational cost, but handling non-dominant individuals is included as part of the archiving technique. With the explained archiving mechanism, these dominance relationships are not taken into account, since it covers Pareto dominance between two individuals which belong

to two different ϵ-glitches. Therefore, computational cost would be wasted in search zones of no interest (those which have a higher number of dominant points than the one already obtained which yields zero segmentation error). To cover this special case, when a new individual is added to the archive with zero representation error, the archive size is reduced to its number of segments.

A final overview of the detailed archiving process can be detailed in the following steps:

- Initialize archive with size n and fill with individuals obtained with applied directed mutations from an individual with either, the highest or the lowest possible number of segments, obtaining one individual for each possible number of segments
- While stopping criterion is not met
  - Select parents
  - Apply transformation operators to parents
  - For each children produced
    - Compare to archive individual with the same number of segments, and update as required
    - If the updated individual has zero representation error, update archive size as required (if the number of segments of the individual is smaller than archive size)
- Output archive results

7.3 Initialization revisited: multiobjective local search techniques

Initialization techniques for the faced domain were presented in section 6.4.2. Several techniques were analyzed, from the default genetic initialization up to the application of specific heuristic local search techniques. Among the issues of this last choice were the difficulty to obtain well distributed Pareto fronts from the error input parameters, the requirement for independent runs and the computational cost, which led to disappointing results (as shown in section 6.5.1).

This section will present alternative, parameter-free versions of two heuristic local search techniques previously introduced, according to their traditional, single objective versions: Top-Down and Bottom-up algorithms. These techniques were originally introduced in section 2.8.2, have been used through chapter 3, and considered as local search techniques in section 6.4.2. A short introduction will be provided for each of the two algorithms, along with a data flow diagram in order to provide the required context for their multi-objective versions.

Top Down algorithm (Ramer, 1972) is an offline process based on finding the best splitting point (understanding by this that measurement which divides the trajectory into the two segments with the lowest added errors) recursively, until all the resulting segments have an error value bellow a user defined boundary. The Top Down algorithm is applied in a wide variety of domains and fields, being also known by different names(Duda & Hart, 1973). Figure 7.11 shows this process.

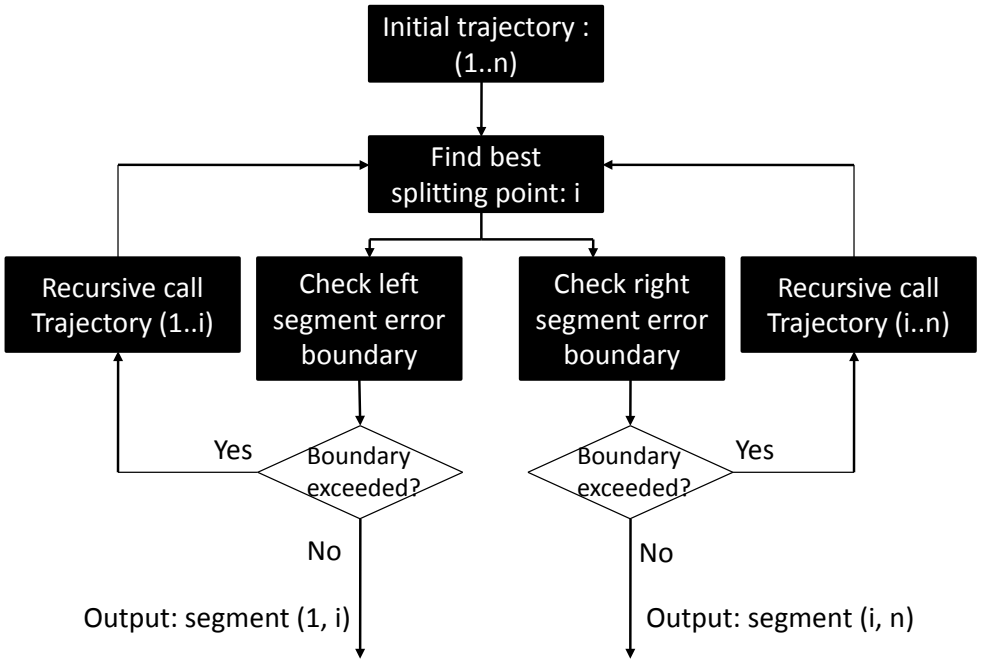


Figure 7.11: Top-Down traditional implementation

The multi-objective version of the Top Down algorithm suppresses the two issues available in the traditional implementation: the recursive calls (which may prevent the application of the algorithm to figures with a large number of points) and the user configuration (which introduces the issues previously described in the obtaining of a whole Pareto front). At each step, the best splitting point is located (the one which provides the smallest representation error), a new individual is generated adding that new dominant point and the costs of the possible segments are updated (implying the recomputation of the costs of the segments from the dominant point immediately to the left of the new splitting point and those from the splitting point to the dominant one immediately to its right).

Each step of this iterative process can be seen as a guided mutation, and, thus, the considerations introduced in section 7.1 can be applied, regarding the update of partial fitness values in the chromosome and the computation of the new fitness.

According to the explained procedure, no recursive calls are included, and each split point choice has a global view of the representation error (as opposed to the partial one available in the traditional implementation). Figure 7.12 represents the multi-objective version implementation of this algorithm.

Bottom up algorithm(Keogh et al., 2003) is an offline process complementary to Top Down, where the time series is initially divided into every possible segment (composed of two measurements) and finds the best possible segment fusion afterwards (understanding by this the fusion which obtains the segment with the lowest error) until any possible fusion obtains a segment having an error above a user defined boundary. The bottom up algorithm, as well, has spread to different fields and research areas using different names, such as the computer graphics domain and decimation methods(Heckbert & Garland, 1997). Figure 7.13 shows this process.

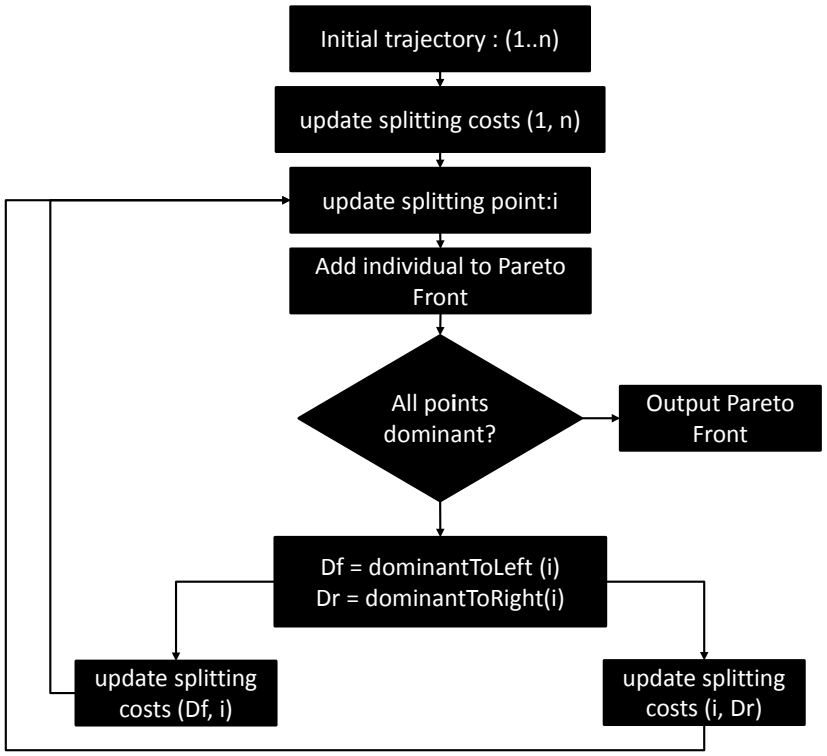


Figure 7.12: Proposed top-down multiobjective implementation

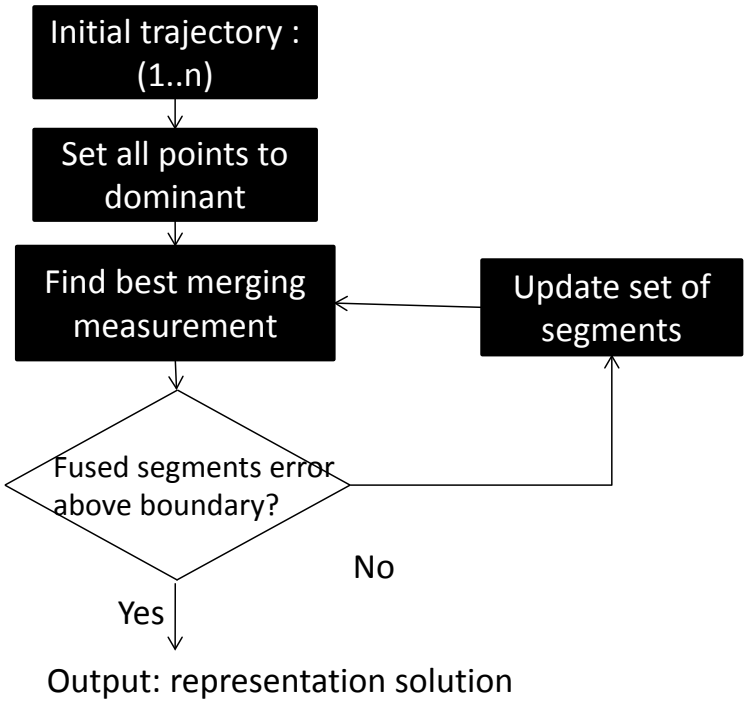


Figure 7.13: Bottom-up traditional implementation

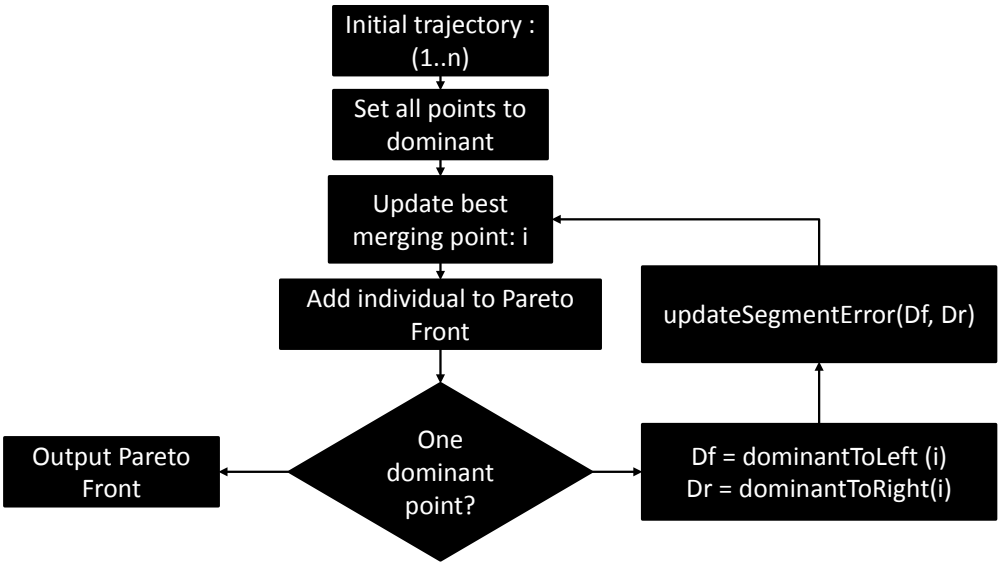


Figure 7.14: Proposed Bottom-up multiobjective implementation

The multi-objective version of bottom-up algorithm removes the user-defined boundaries for the algorithm termination, being this ending triggered once no further merging can be performed. Figure 7.14 presents the multi-objective version. It must be noted that each update here triggers only one segment update, while every new splitting point in the top down algorithm triggered the recomputation of all the possible new splitting points for the two new segments created in the representation. Once again, the efficient fitness computation from section 7.1 can be applied to reduce the computational cost.

An additional advantage presented by these multiobjective local search approaches is related to the archiving analysis introduced in section 7.2. The presented archiving technique reduces its archive size once a zero error individual has been found with a given set of dominant points lower that the number of points of the problem. The multiobjective bottom-up technique starts, precisely, looking for zero error individuals (by merging the whole curve at the points of the least error increase) and may provide a reduced archive size to the algorithm and thus focus the search.

7.4 Stopping criterion

Chapter 5 presented a general stopping criterion for MOEAs, based on the principles of quality assessment presented in section 2.6. The difficulty regarding quality assessment in multi-objective problems was, as analyzed, that several measures have to be considered jointly: closeness to the True Pareto Front, spread of the front,... This is what lead to the proposal of progress indicators (quality indicators modified to measure the improvements per generation) in section 5.2.1 and, finally, along with a stagnation detection procedure based on the analysis of the gathered data, propose the LSSC stopping criterion (section 5.3).

Application to polygonal approximation, as presented in chapter 6, has an important difference to the general case: the maximum number of individuals in the True Pareto Front

Table 7.1: Efficient Multi-objective evolutionary segmentation algorithm summary

Parameter	Description
MOEA algorithm	Own
Representation	Binary vector + additional fitness information
Objective Functions	2, dominant points and Integral Squared Error
Initialization process	Uniform with local search boundaries
Crossover operator	1-point crossover fitness-aware
Mutation operator	bit-flip mutation fitness-aware
Crossover probability	0.9
Mutation probability	$1/n$
Population size	- (not applicable to algorithm)
Generation number	- (stopping criterion)
Archive size	- (established after initialization)
Archive procedure	Epsilon glitches
Stop. crit. window size	30

can be calculated before the application of the algorithm, and the archive size (and thus, the obtained fronts from it) has been configured accordingly. The implications of these characteristics greatly simplify the quality assessment problem, which is now reduced to a closeness comparison.

This simplification can be translated to a relevant progress indicator, which simply measures the improvements at the different possible individuals of the front between different generations, providing a simple boolean indicator value (the front has improved at least one of its individuals or not) rather than a quantitative numeric value. The data gathering process, according to this new boolean assessment value, is also simplified: this boolean assessment values are stored in a certain window of values, and the processing is limited to determining whether there have been any improvements over any of the considered generations.

This approach is similar to the one presented in section 7.2 as an alternative archiving technique. In fact, the implementation of this stopping criterion can be introduced into the archiving procedure introducing a simple operation: during the archiving procedure, if any individual is modified, that generation's boolean assessment value is marked as true. This can be performed in such a simple way since the archive using this technique is itself the population we need to test the stopping criterion.

There are several advantages to this procedure: the parameters are reduced (only the window size is still required), there is no need for the computation of quality indicators and the computation of the new assessment indicator can be performed at a very low cost (using the previous SPEA2 approach) or with no cost (if the alternative archiving is used).

7.5 Final proposal summary

The final proposed algorithm has the configuration presented in table 7.1

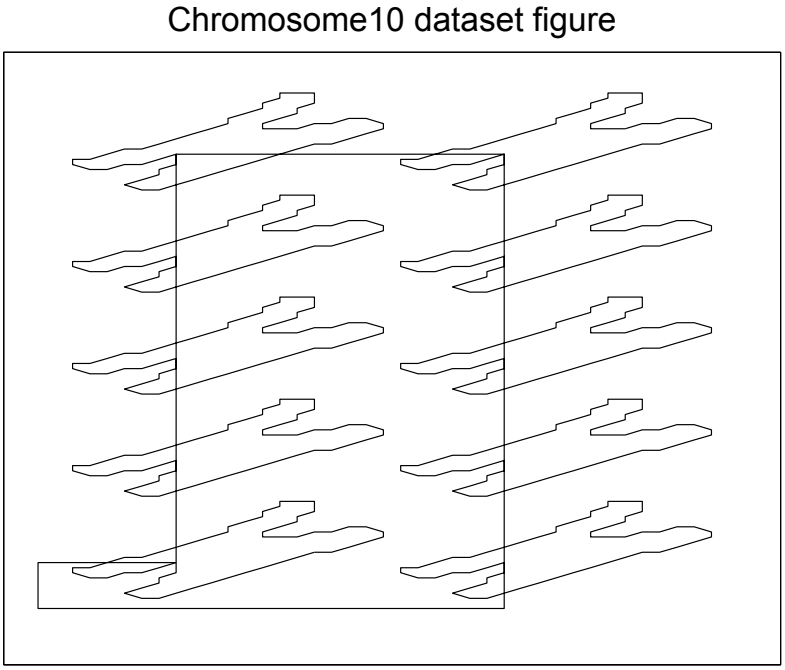


Figure 7.15: Example of ten chromosome curves linked together by the designed replicating mechanism to provide problem instances with increasing difficulty

7.6 Experimental results

7.6.1 Dataset used

This experimental section has to deal with different issues, since several proposals have been presented to deal with the existing issues in the approach presented in chapter 6. Since the final objective is to obtain results of similar quality to the ones presented in table 6.8 at a lower computational cost, this quality comparison must be run on the same dataset (which is composed of the three curves presented in figures 6.11-6.13.

However, to test the improvements provided by some the introduced techniques, this chosen dataset may not provide enough complexity (in terms of number of points in its figures). There are other alternatives in the literature which are used by different authors along with their presented techniques, but there is not (to the best of our knowledge) a dataset or curve generator which provide problem instances of increasing complexity (where that complexity can be controlled by the researcher). To solve this issue a very simple replicating mechanism has been introduced in the following results.

The created replicating mechanism takes three parameters: the desired curve, the number of total copies and the number of copies per row. Aferwards it generates a matrix of linked copies of the curve, where as many copies per row as indicated are included, and provides a final closed curve. Figure 7.15 shows an example of ten copies of the chromosome, with five copies per row.

7.6.2 Fitness-aware operators experimental results

The improvements presented in sections 7.2 and 7.3 are based or caused by the initial introduction of fitness-aware operators into the presented algorithm. Thus, these results will be the first ones examined, in order to provide a solid basis for the following experiments.

In section 7.1, two different and complimentary considerations were presented, and they are reflected into current experimental results. The first alternative is based on two fitness-aware operators which, according to an alternative chromosome representation including partial fitness information, are able to perform only partial fitness updates along with the proper transformation operators. The second alternative adds to the fitness-aware operators the modification of the Pseudorandom number generator of the mutation operator, in order to reduce its computational cost.

Two different comparison procedures will be carried out: the first process will compare the running time of the different alternatives, while the second will compare the quality of the final results obtained. The objective of this comparative procedure is to determine the improvement obtained (in terms of computational cost) with the presented alternatives and provide assessment over whether this improvement implies degradation over the quality of the results obtained. Fitness-aware operators (used in both alternatives) only imply a change in the fitness computation, but not its result, such that no quality comparison is required. However, changing the pseudorandom number generator procedure (used in the second alternative) does provide different final results (due to the stochastic nature of genetic algorithms), so a statistical comparison process will be carried out over the quality of the results.

The running time comparison will be based on 50000 individuals (equally divided into different runs with crossover probabilities 0.1, 0.3, 0.5, 0.7 and 0.9, each with 10000 individuals) for each of the considered test problems, the application of crossover and mutation operators, and, finally, the recalculation of the fitness value, measuring the complete running time. Two different segment approximation techniques will be used, the linear approximation of the dominant points, and a more complex one based on the least squares approximation of all the points in the segment (which was the approach used in chapter 3). This last least squares approximation is not the one followed by polygonal approximation techniques, but provides and in-depth overview of the behavior of the presented technique as the fitness function becomes harder to be computed.

It must be highlighted again that this running time comparison does not perform the whole evolutionary cycle, focused only on the transformation operators and their associated fitness updates. For the quality comparison results, thirty runs of the algorithm as presented in table 6.3 will be run, introducing the presented transformation operators, and the results will be compared according to the proper statistical testing.

Comparisons ver different evolutionary approaches to a problem are usually measured in terms of function evaluations. In this case such a comparison cannot be performed, since the proposed approaches use no explicit function evaluation (or, at least, this function evaluation is not comparable to the original one), since fitness-aware operators recompute the new fitness values as an integral part of their procedure. For this reason, the results are compared according to their running time.

Five different configurations have been used as well: traditional figure, 10 copies of the traditional figure in 2 columns, 50 copies in 10 columns, 100 copies in 10 columns and finally 500 copies in 50 columns (the procedure to obtain these curves has been explained in section

Table 7.2: Fraction of the original running time achieved by the proposed techniques for the chromosome curve problem instances

Configuration	Problem instance size				
	Original	10	50	100	500
Fitness-aware Linear	0.86	0.74	0.74	0.73	0.75
Fitness-aware Least Squares	0.80	0.66	0.66	0.65	0.67
Reduced Mutation Linear	0.61	0.48	0.47	0.46	0.46
Reduced Mutation Least Squares	0.51	0.40	0.44	0.43	0.41

Table 7.3: Fraction of the original running time achieved by the proposed techniques for the semicircle curve problem instances

Configuration	Problem instance size				
	Original	10	50	100	500
Fitness-aware Linear	0.74	0.73	0.74	0.70	0.72
Fitness-aware Least Squares	0.72	0.65	0.62	0.65	0.66
Reduced Mutation Linear	0.48	0.47	0.47	0.45	0.45
Reduced Mutation Least Squares	0.44	0.42	0.38	0.43	0.39

7.6.1).

Tables 7.2-7.4 represent the fraction of the original approach's time taken by each of the configurations introduced, being presented as a graphical comparison in figure 7.16

Running time results show that the approach based only on fitness-aware operators requires around 70% of the original computational cost, while the reduced PseudoRandom generation mutation lowers this value down to a 40% of the original time.

Quality comparison tests over the hypervolume indicator values of the final Pareto fronts obtained a p-value for the used normality test of 6E-16, implying that the analyzed data did not follow a normal distribution. Thus, the Wilcoxon test was applied over the results, obtaining a p-value of 0.54841, determining that the data cannot be considered to come from distributions with a different median. This proves that the introduced pseudorandom number generation system does not hamper the final quality of the results, while providing additional saves in the computational cost of the algorithm (as seen in tables 7.2-7.4).

Table 7.4: Fraction of the original running time achieved by the proposed techniques for the leaf curve problem instances

Configuration	Problem instance size				
	Original	10	50	100	500
Fitness-aware Linear	0.80	0.72	0.69	0.77	0.82
Fitness-aware Least Squares	0.75	0.67	0.70	0.71	0.83
Reduced Mutation Linear	0.44	0.46	0.48	0.48	0.53
Reduced Mutation Least Squares	0.41	0.39	0.39	0.40	0.46

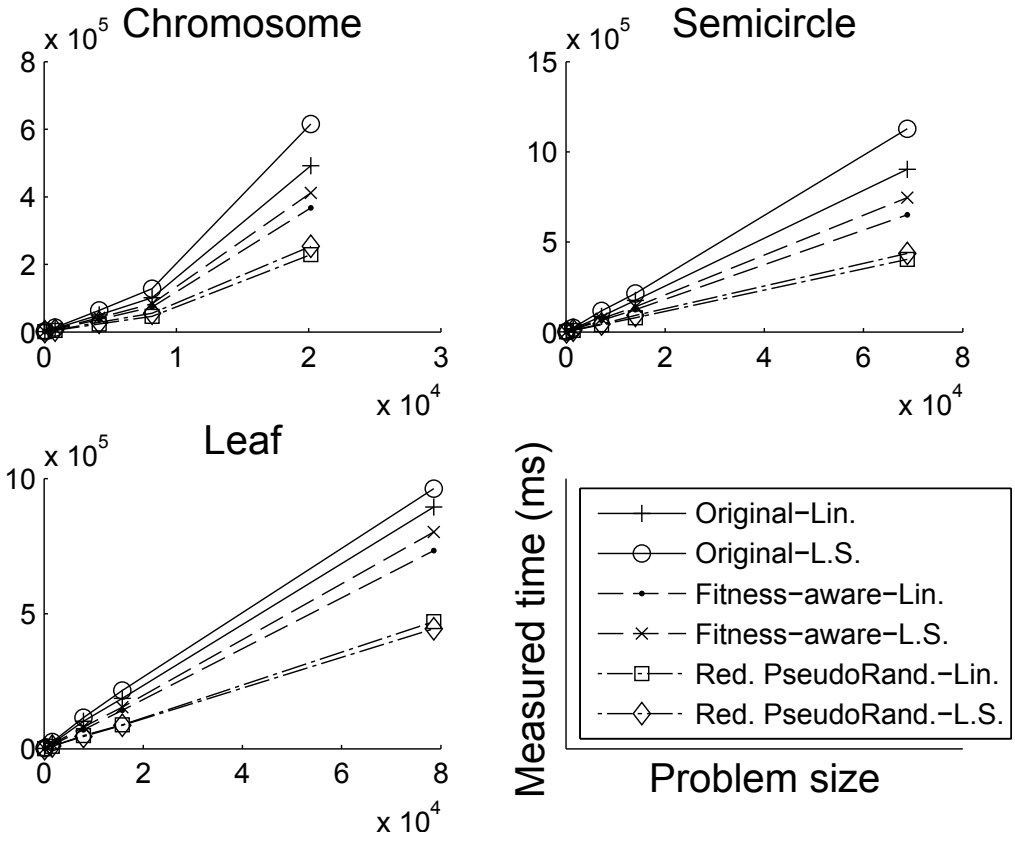


Figure 7.16: Comparison of the running times provided by the presented fitness aware transformation operators on the chosen dataset

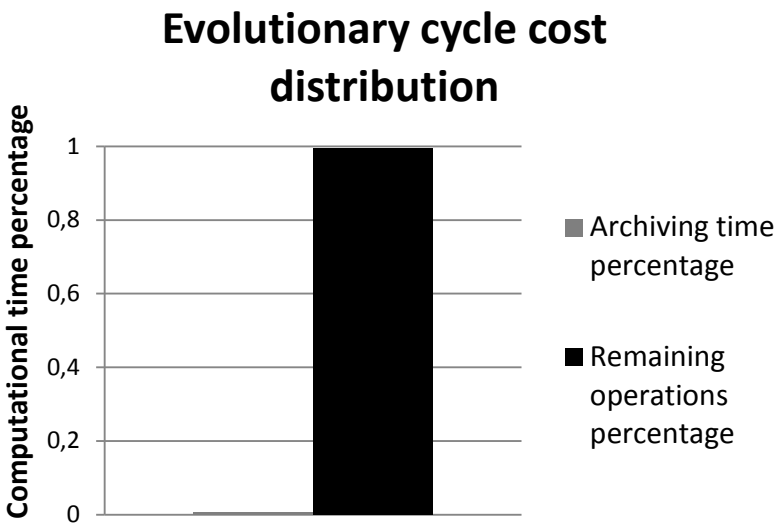


Figure 7.17: Computational cost distribution between archiving technique and the remaining procedures of an evolutionary cycle after the introduction of the designed archiving procedure

7.6.3 Alternative archiving procedure

The first relevant comparison, according to the motivations presented in section 7.2.2, is the computational time which is now used by the archiving procedure. The original results for the SPEA2 technique have been presented in figure 7.7. The results with the presented technique are shown in figure 7.17. These results are general for all the different figures in the dataset. As this figure shows, the percentages have been swapped, spending now more than 99% percent of the available computational time in the evolutionary search instead of the archiving technique, and thus providing a much better focus for the computational cost.

If the population size is too large or the number of generations too the effect of the saved computational cost may not be clearly measurable, since the stagnation will have occurred before the actual stop of the algorithm. Thus, population size and number of generations have been reduced from the values presented in table 6.3. The chosen population size used is 200, and the algorithm is left to run for 200 generations. The running time used for each of this independent executions is measured, and afterwards thirty different runs of the proposed technique are performed, each of them according to an individual previously measured running time as its stopping criterion. The hypervolume indicator values are built afterwards according to these results and the statistical significance of the differences are tested according to Wilcoxon statistical testing (none of the results were normally distributed) with a 95% confidence interval. The results of these procedures are shown in table 7.5. Figure 7.18 summarizes these results.

The results for the three initial figures do not show statistically significant differences between the two techniques, while the results for the three harder ones are clearly dominated by the epsilon-glitches based technique. The explanation for this fact is clear: when the final solution can be easily reached, the improved distribution of the solutions provided by the environmental selection technique allows SPEA2 to obtain solutions of similar quality, even though the computational effort spent in the proper search process is smaller (as seen in the comparison of Figures 7.7 and 7.17). As the problem instances become harder and

Table 7.5: Final Pareto front hypervolume comparisons introducing the novel archiving technique

Curve	Epsilon Hyp. results Mean	Std	SPEA2 Hyp. results Mean	Std	Best
chrom	0.99000	0.00006	0.99001	0.00005	-
leaf	0.99533	0.00001	0.99532	0.00001	-
semi	0.99409	0.00008	0.99412	0.00003	-
chrom10	0.99924	0.00002	0.99858	0.00093	eps.
leaf10	0.99961	0.00001	0.99812	0.00139	eps.
semi10	0.99944	0.00002	0.99814	0.00175	eps.

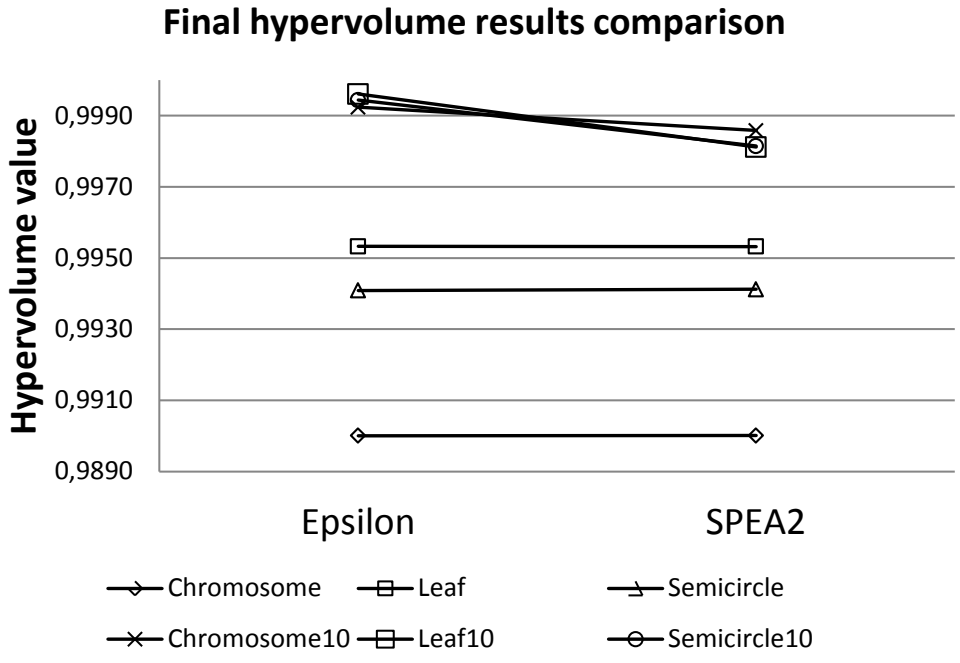


Figure 7.18: Final Pareto Front hypervolume results comparison applying the presented archiving technique

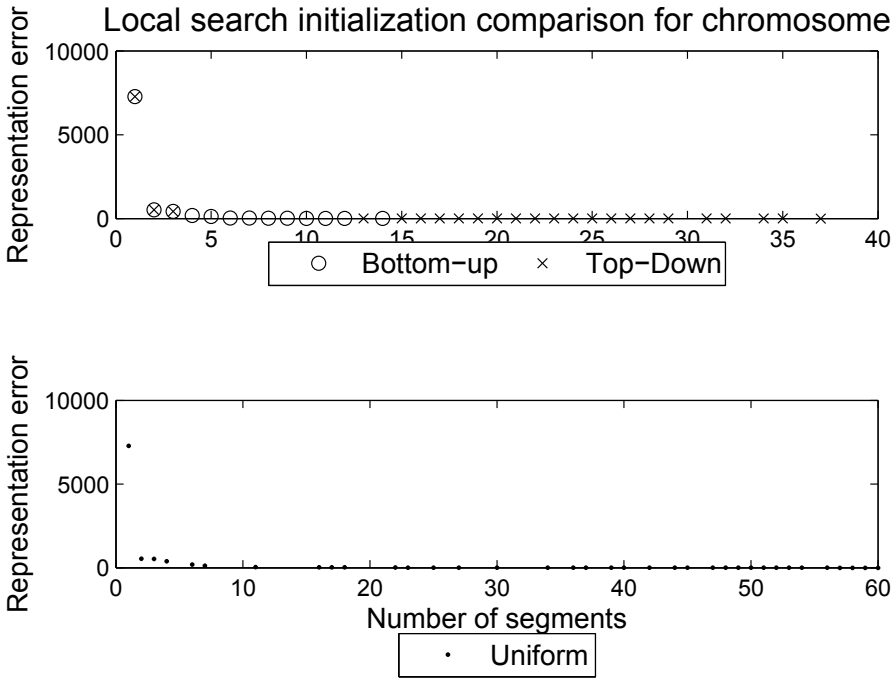


Figure 7.19: Chromosome initialization comparison

the extent of search required to reach a reasonable Pareto front grows, the focus on the search process of the epsilon-glitches technique pays off for the poorer solution distribution, providing it with substantial better results in terms of final hypervolume.

7.6.4 Multiobjective local search initialization

It is interesting to notice, as explained in section 7.3, the complementary nature of the two multi-objective techniques presented, since one applies its heuristic with a value of 1 dominant point and applies successive splitting over the figure (Top-Down) and the other begins with a solution with all of its points considered dominant and applies successive merging (Bottom-up). Since the solutions tend to degrade with the successive application of the heuristic, each of them will be more successful at their initial individuals.

Three different comparisons of the two multi-objective local search techniques and the original uniform approach for the three different curves in the dataset are presented in figures 7.19-7.21. The only individuals included are those non-dominated (the Pareto fronts for the three techniques). Regarding the previously stated complementary nature of the local search processes, it can be clearly observed in these figures.

The results for the four techniques, including their mean and median values for the hypervolume of the obtained Pareto fronts are included in tables 7.6 (initial front values) and 7.7(final front values). Also, a best technique column has been added. This value is calculated according to a Wilcoxon test with a 95% confidence performed over 30 different executions, since the values do not follow a normal distribution (according to a Shapiro-Wilk test). If one technique is superior to the remaining ones, its name is included, otherwise the '-' value is included.

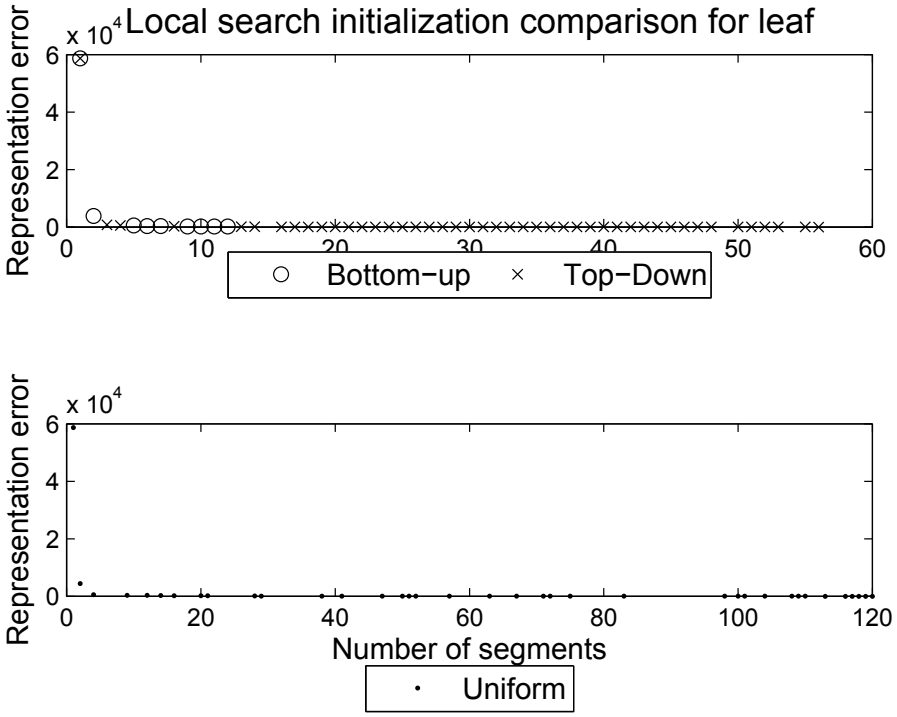


Figure 7.20: Leaf initialization comparison

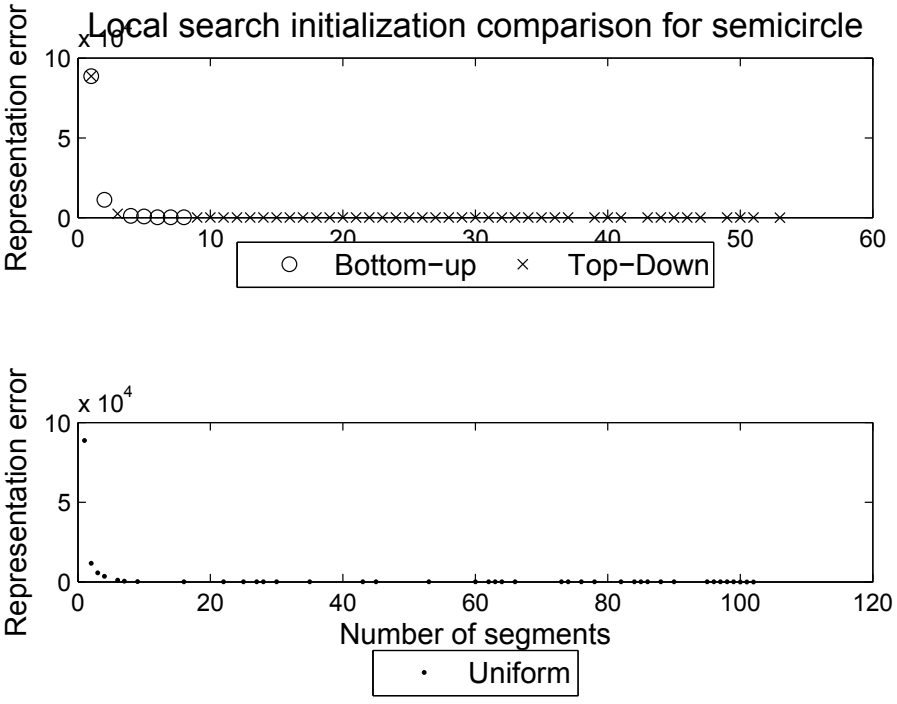


Figure 7.21: Semicircle initialization comparison

Table 7.6: Initial populations comparison

Figure	Bottom-up		Top-down		Local search		Uniform		Best
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	
Chrom	0 98647	0 98647	0 98646	0 98646	0 98651	0 98651	0 98436	0 98427	L S
Leaf	0 99355	0 99355	0 99322	0 99322	0 99365	0 99365	0 99271	0 99281	L S
Semi	0 99157	0 99157	0 99183	0 99183	0 99218	0 99218	0 99101	0 99111	L S

Table 7.7: Final populations comparison

Figure	Bottom-up		Top-down		Local search		Uniform		Best
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	
Chrom	0 98665	0 98664	0 98667	0 98671	0 98665	0 98667	0 98671	0 98672	Unif
Leaf	0 99376	0 99376	0 99374	0 99376	0 99376	0 99376	0 99377	0 99378	-
Semi	0 99206	0 99219	0 99213	0 99217	0 99219	0 99219	0 99213	0 99217	L S

Regarding the initial populations, the local search techniques are able to find the individual with zero error with a much lower number of segments that the uniform approach. This is especially important since finding solutions with a higher number of segments does not provide information to the final solution, and can be considered a waste of computational cost. Also, this information could be used to manage the size of the archive, allowing a reduction of the computational cost. The representation errors for the individuals for the different number of segments are also clearly better that those obtained by the uniform initialization, which is reflected in the results in table 7.6.

In the analysis of the final populations results, different cases appear. For easy problems, such as chromosome, the uniform initialization provides better final results, while as the problem difficulty is increased, the statistical difference first disappears in leaf curve and finally the local search initialization provides better results in the hardest problem, the semicircle.

The analysis of these results can be obtained from the previous remark on initial populations: the repeated application of a heuristic approach provides an ever growing error (as seen in the comparisons of the the local search approaches in figures 7.19-7.21). Translated to the evolutionary approach, the local search initialization introduces a certain bias to the search performed by the evolutionary algorithm, according to its underlying heuristic. Even though the initial results are clearly improved, the final ones are too guided by this heuristic, and thus, they fall into local minima solutions. To highlight this analysis and provide a further understanding to the presented techniques, figures 7.22-7.24 provide a comparison of the evolution of the hypervolume value through the different generations of the algorithm.

The presented results seem to point to a combination of both techniques to provide initial populations that, while benefiting from the enhanced initial populations of local search techniques, are not hampered by the heuristic focus. Also, an initial run of constructive techniques such as bottom-up can be used for the configuration of some algorithm parameters like archive size, an application which can be efficiently combined with the archiving procedure from section 7.2.

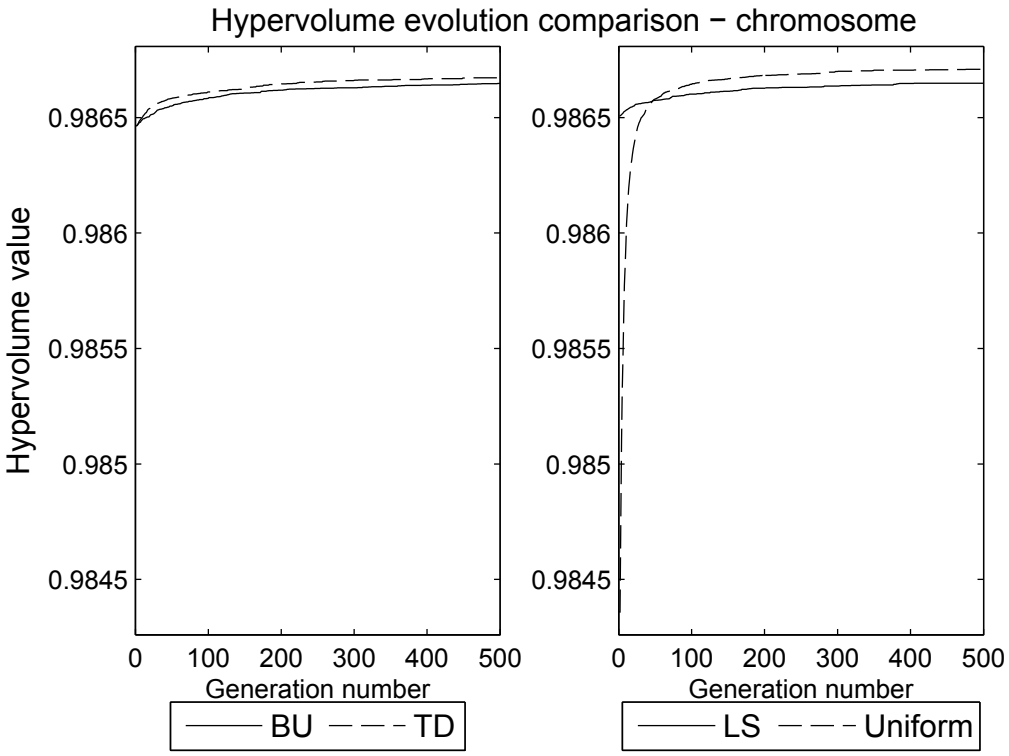


Figure 7.22: Hypervolume evolution comparison

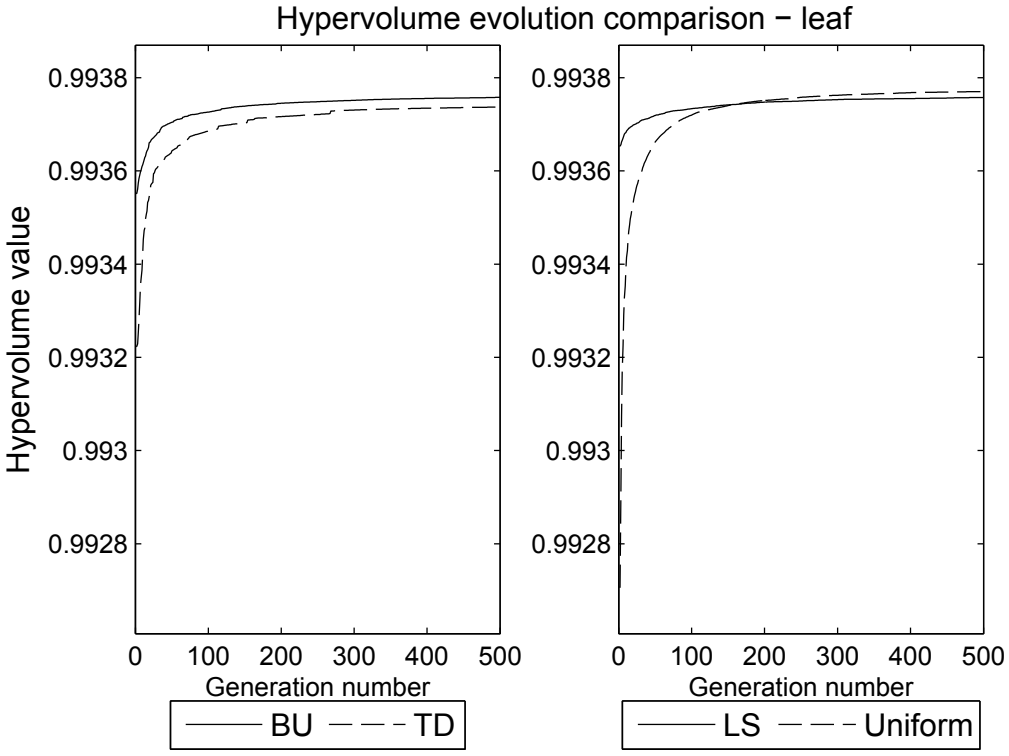


Figure 7.23: Leaf evolution comparison

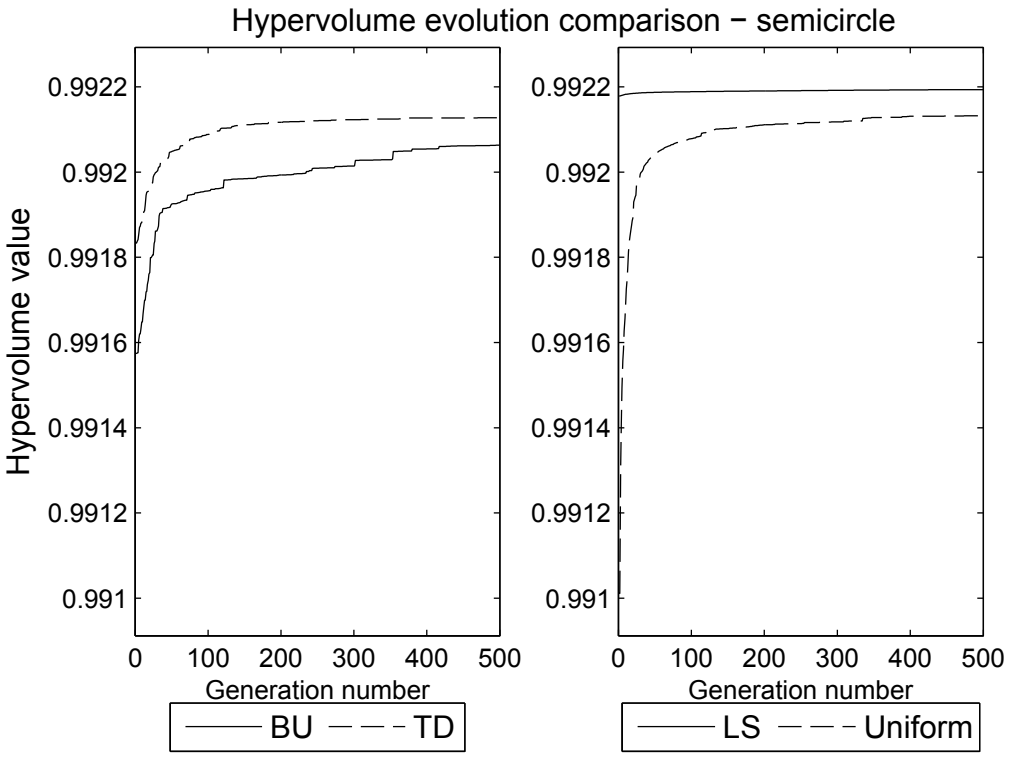


Figure 7.24: Semicircle evolution comparison

Table 7.8: Stopping generation comparison

Curve	Stopping generation		
	Mean	Median	Std
Chromosome	144.1	133	40.66
Leaf	383.3	359	104.3
Semicircle	291.87	271.5	86.29

7.6.5 Stopping criterion

The results for the three different curves in the dataset, as presented in figures 7.19-7.21 have been used to test the efficiency of the stopping criterion. The only parameter required for the stopping criterion, as previously explained, is the window size, which, according to section 5.3, has been set to 30 generations. The results regarding the stopping generation are presented in table 7.8, while the final hypervolume results are presented in table 7.9. Figures 7.25-7.27 show examples of stopping generations for the three curves in the dataset.

The results of the stopping generations are consequent with the problem difficulty, and the hypervolume comparison shows that this reduction in the number of generations does imply a lower final hypervolume value (as suggested by the a-posteriori configuration which had been carried out in chapter 5, which lead to the choice of 2000 generations parameter).

A stopping criterion is a trade off between computational cost and solution quality, and also a requirement for problem instances where the complexity or the approximate

Table 7.9: Final hypervolume values comparison

Curve	Default		Stop crit		Difference	
	Mean	Median	Mean	Median	Mean	Median
Chromosome	0 9867657	0 9867616	0 9867222	0 9867248	4 35187E-05	3 67643E-05
Leaf	0 9938007	0 9938015	0 9937925	0 9937926	8 2155E-06	7 36015E-06
Semicircle	0 9922071	0 9922077	0 9921943	0 9921953	1 27653E-05	1 20117E-05

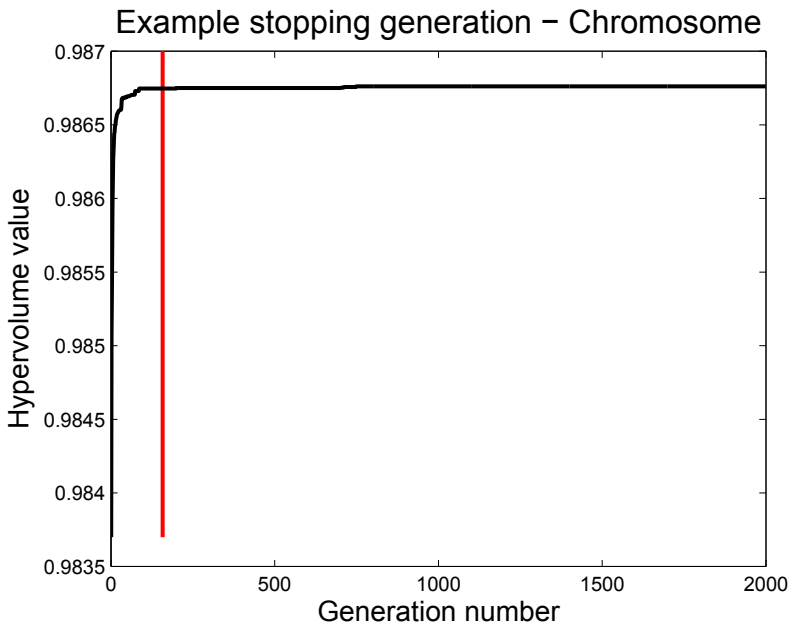


Figure 7.25: Stopping criterion application example to the Chromosome curve

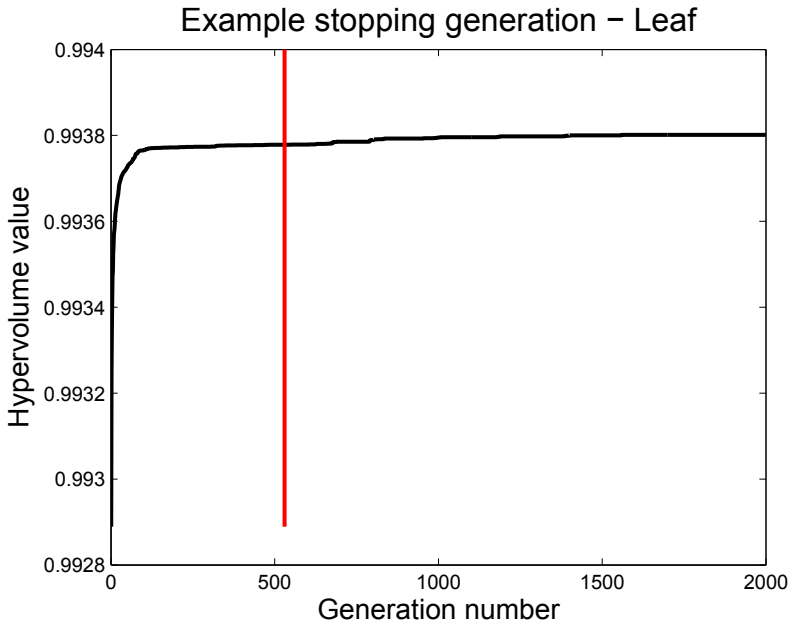


Figure 7.26: Stopping criterion application example to the Leaf curve

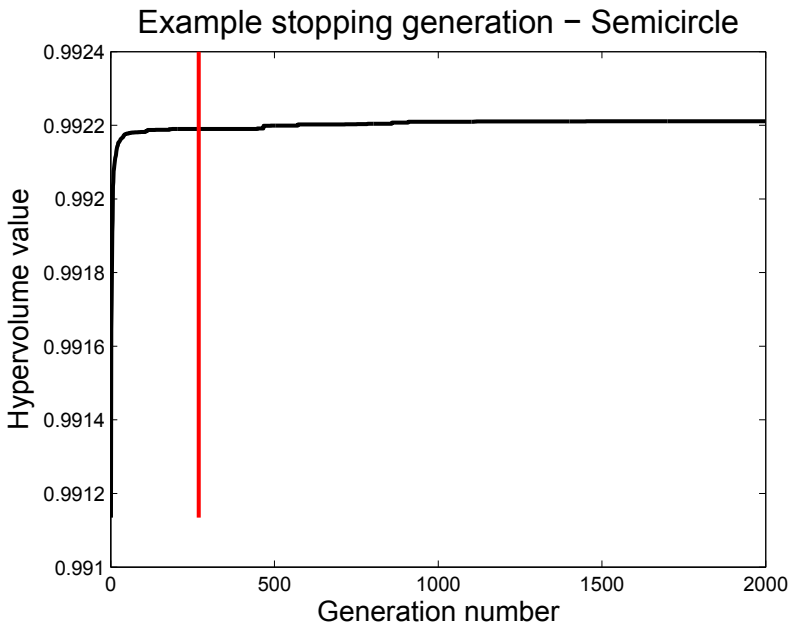


Figure 7.27: Stopping criterion application example to the Semicircle curve

computational cost is unknown a-priori. Even though for simple problem instances choosing a high fixed computational budget may yield better results, such an approach is inapplicable for real problem instances.

7.6.6 Final proposal results

The configuration for the final proposal, according to the presented results, was detailed in table 7.1. The different improvements detailed through current chapter have been introduced into this final proposal, even if some only in a partial approach, such as multiobjective local search initialization, used to provide the boundaries to the uniform initialization procedures, which provides a richer diversity, as analyzed in section 7.6.4. The results for each of the individuals in the three main figures in the dataset are presented in table 7.10. A detail of this comparison is presented in figure 7.28.

The comparison to the traditional techniques, even though the algorithm performs worse than SPEA2 version with fixed number of generations (as presented in table 7.10), produce exactly the same results of terms of statistical best technique as the ones presented in section 6.5.2, particularly in tables 6.10-6.13.

The measured speedup versus the original proposal varies among the different problem instances, since complexity of the archiving procedure of SPEA2 depends on the population and archive size, which is set by the problem, and also the problem characteristics affect the stopping criterion. For the leaf curve, the speedup obtained is around 38, for the chromosome curve, the speedup is around 68, and for the semicircle, 55. This speedup gains relevance as the problem complexity grows. For the leaf10 curve, the running time of the original algorithm on the test computer is around 29 hours, without any real parallelization possibilities, since the archiving must be centralized. The speedup measured in this case, forcing the proposal to run for the same 2000 generations, is around 653. These results are represented gra'hically

Table 7.10: Pareto Front dominant points / integral squared error results for the dataset

Dom points	Chromosome		Semicircle		Leaf	
	Mean	Std	Mean	Std	Mean	Std
1	7280	0	88648	0	58661	0
2	520	0	11589 33	735 71	3830 78	49 27
3	382 07	16 32	2880 27	423 09	518 34	52 33
4	156 22	22 77	1183 73	93 45	410 49	40 96
5	106 84	7 62	593 75	47	329 02	20 7
6	31 31	11 97	158 66	17 05	260 47	26 35
7	21 49	6 36	127	11 26	209 72	17 68
8	14 57	3 13	99 55	8 42	178 5	14 41
9	12 72	1 8	81 15	7 67	154 41	12 71
10	8 8	1 75	65 32	3 51	123 68	12 68
11	7 59	1 19	49 78	7 62	104 05	9 25
12	6 19	0 74	37 63	5 54	78 31	13 01
13	5 19	0 66	27 66	3 86	60 72	11 04
14	4 53	0 56	21 16	2 93	49 76	6 97
15	3 99	0 42	16 76	2 01	39 82	5 76
16	3 62	0 33	14 47	1 17	31 19	4 42
17	3 28	0 3	12 94	0 56	25 85	4 07
18	2 96	0 2	11 87	0 45	20 72	3 91
19	2 67	0 18	10 61	0 42	17 7	3 06
20	2 44	0 14	9 59	0 28	15 52	2 49
21	2 24	0 13	8 61	0 34	13 49	1 78
22	2 02	0 1	7 6	0 39	11 89	1 67
23	1 82	0 1	6 7	0 39	10 66	1 6
24	1 62	0 09	5 96	0 41	9 69	1 37
25	1 42	0 09	5 37	0 41	8 81	1 11
26	1 23	0 08	4 84	0 33	8 07	0 96
27	1 08	0 06	4 46	0 29	7 35	0 84
28	0 96	0 05	4 09	0 27	6 72	0 73
29	0 82	0 03	3 73	0 27	6 16	0 65
30	0 72	0 04	3 38	0 26	5 64	0 57
31	0 59	0 03	3 04	0 26	5 16	0 49
32	0 5	0 03	2 73	0 23	4 74	0 42
33	0 37	0 02	2 51	0 19	4 33	0 35
34	0 31	0 01	2 33	0 16	3 96	0 3
35	0 17	0 02	2 15	0 15	3 65	0 23
36	0 15	0	1 98	0 14	3 38	0 19
37	0	0	1 79	0 14	3 11	0 17
38			1 63	0 14	2 85	0 16
39			1 45	0 13	2 6	0 16
40			1 28	0 13	2 36	0 15
41			1 14	0 11	2 15	0 14
42			1 01	0 1	1 95	0 14
43			0 88	0 06	1 75	0 13
44			0 8	0 05	1 56	0 13
45			0 65	0 06	1 39	0 12
46			0 62	0 02	1 21	0 11
47			0 47	0 02	1 04	0 1
48			0 46	0	0 89	0 09
49			0 31	0	0 74	0 07
50			0 31	0	0 6	0 07
51			0 15	0	0 49	0 04
52			0 15	0	0 35	0 04
53			0	0	0 3	0 02
54					0 16	0 01
55					0 15	0
56					0	0

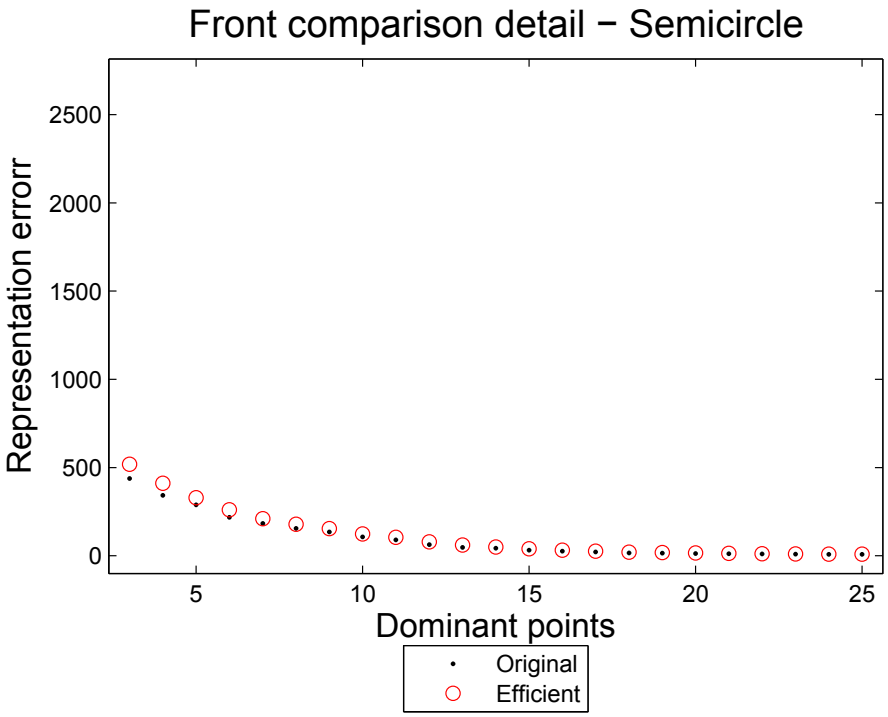


Figure 7.28: Detail comparison of the results between the original and the efficient MOEA approaches for the Semicircle curve

in figure 7.29

While SPEA2 proposal clearly lacked parallelization possibilities (99% of the time was spent in the archiving procedure, as shown in figure 7.7), the introduction of the Epsilon glitches archiving procedure introduces clear parallelization opportunities, since the archiving procedure can be easily synchronized at certain intervals at a low computational cost, thus allowing for an even better speedup for hard problem instances.

7.7 Conclusions

This chapter has dealt with one of the general objectives which had been established from the thesis title: how to adapt a general multiobjective evolutionary approach to a practical problem. Chapter 6 introduced the MOEA approach and tested its validity versus a series of heuristic and metaheuristic alternatives, according to a general algorithm and its associated operators. The proposal through that chapter was how to establish a metaheuristic which did not have such a high problem dependence as the heuristics reviewed (or even the one designed in chapter 3). This chapter takes the complemetary approach: analyze the different steps of the evolutionary cycle and adapt them as required to the problem domain.

Different proposals have been presented and tested individually to provide the final evolutionary procedure: representation and fitness-aware operators according to it, an alternative archiving procedure, the introduction of multiobjective local search operators and their use for initialization of the population and, finally, the introduction of a stopping criterion. Also, a new mechanism to generate arbitrarily harder problem instances according to the three

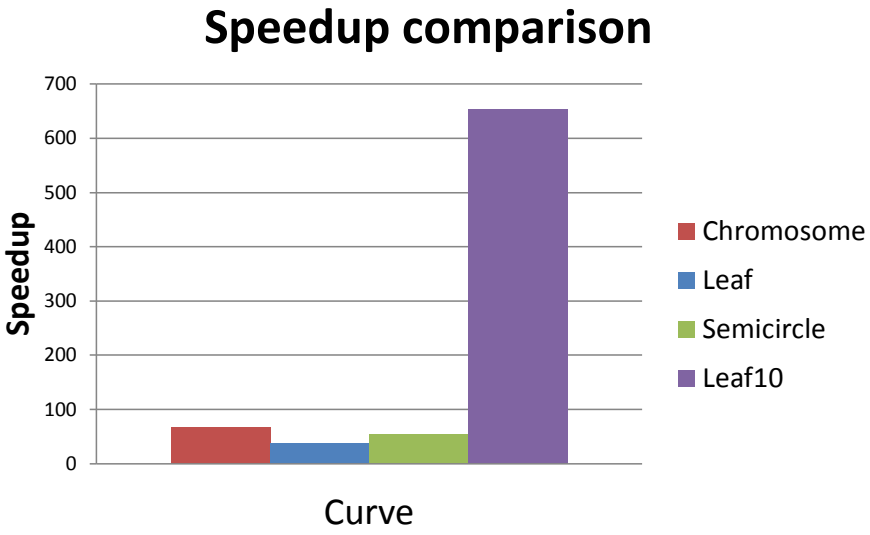


Figure 7.29: Speedup comparison between original and efficient MOEA approaches

basic curves has been developed.

The new chromosome representation presented introduces into it a codification of partial fitness information. This representation allows the formulation of fitness-aware crossover and mutation operators which perform partial fitness updates according to the codified fitness information already in the chromosome. Two different alternatives are presented according to these operators, one of them keeping the original formulation of the operators and completing it with the partial fitness updates and an additional one with a modified Pseudorandom number generator which enhances the computational time of the mutation operator. These alternatives are tested according to a proposed difficulty scaling procedure, based on the creation of new closed curves based on the repetition of a given shape. The results show that the proposed improvements over the traditional evolutionary approach manage to reduce the computational time down to around a 40% of the original time without any statistically significant degradation in the quality of the obtained solutions.

Traditionally, the most expensive process in an evolutionary algorithm (mainly for single objective approaches) was the fitness computation. After the introduction of the enhanced representation and fitness-aware operators, the overall cost distribution was analyzed, highlighting that the archiving was taking almost the whole time of every evolutionary cycle, a fact derived from the complexity order of the environmental selection used by the procedure. A new archiving process, established according to the bi-objective nature of the problem, being one of these objectives discrete, has been presented. This process only checks dominance within the discrete objective value. The cost distribution after this archiving technique is introduced shows that the computational effort of the algorithm becomes focused on the general evolutionary processes, and it also introduces parallelization opportunities to the resultant algorithm, making it applicable to a wider range of harder problem instances.

Depending on the problem difficulty and the archive and population sizes, this archiving technique may lead to worse results. For the original curves, with 200 individuals and 200 generations, both techniques yield similar results. If those values are increased to the ones

presented in chapter 6, 500 individuals and 2000 generations, the SPEA2 technique does provide better results. The proposed archiving obtains better results with any of the two configurations when harder problem instances are faced.

Local search initialization had been reviewed and discarded in sections 6.4.2 and 6.5.1. The local search procedure analyzed was single-objective, which made the obtaining of specific individuals of the Pareto front hard and computationally costly. Multiobjective versions of two representative constructive and destructive local search techniques, namely Bottom-up and Top-down, have been modified to provide a multi-objective approach with the required characteristics presented. These techniques are embedded as the initialization procedure attempting to benefit from the fast heuristic approach and the thorough metaheuristic search. The obtained results show that the multi-objective techniques are successful in providing statistically better initial populations, however, the final results may be too focused on the heuristic used in these techniques (the provided fronts have a poorer diversity), which makes the evolutionary search less effective, making the results fall into local minima and providing worse final hypervolume values. These techniques have been introduced in the final proposal as a configuration technique for the archive size (reducing it to the lowest number of dominant points required by a zero error solution), however they cannot be discarded as a complete initialization procedure for real case scenarios.

The stopping criterion has been reviewed as well to be applied to the domain. The archive size is configured such that it can contain the whole True Pareto Front, according to the problem specifications. This characteristic simplifies the quality measures (only closeness has to be considered) and, thus, a simpler progress indicator can be proposed and applied to the stopping criterion. A boolean progress indicator has been introduced, determining whether any compression level has been improved during the analyzed generation, and the evolutionary process is stopped after a certain window of generations without any progress in any of the individuals. This progress indicator can also be introduced seamlessly into the novel archiving procedure.

The overall integration of these features into a MOEA proposal has obtained an algorithm which is still able to obtain better statistical results than the set of heuristic alternatives, also managing to scale well with the problem difficulty (which was an issue with the original proposal). The speedup measured for hard problem instances, such as the leaf10 curve, is around 653 times, highlighting the improved applicability of the technique.

8

Conclusions and future lines

> *When you find your path, you must not be afraid You need to have sufficient courage to make mistakes Disappointment, defeat, and despair are the tools God uses to show us the way [ ] Nothing in the world is ever completely wrong Even a stopped clock is right twice a day*

Paulo Coelho, *Brida*, 1990

In the introductory chapter of this work a series of objectives were set and have been covered accordingly. This chapter will provide the assessment over the completion of those objectives, along with the new opened research lines derived from this thesis.

8.1 Conclusions

The application domain has been introduced by means of a practical application to a segmentation problem using data from the Air Traffic Control Domain. This domain exhibits specific characteristics which traditional techniques have difficulties dealing with: presence of noise, long uniform segments, several known movement models, introduction of specific domain based knowledge... A hybrid technique has been developed to deal with these issues. The designed technique works following two phases: the first phase identifies uniform sections over the original data, performing a pre-segmentation of these zones and providing isolated sections of the data to the second technique. The model used in this first phase includes the specified domain knowledge: noise regarding measuring devices, motion information and provides non-uniform sections to the second technique. The second technique is a bottom-up algorithm obtained from the segmentation literature, where the guiding heuristic has been modified to cope with the noisy data.

This initial development presents several keys for the development of the remaining research focuses of the thesis. First of all, it is costly to adapt available techniques to different domains, even though the core knowledge required is similar for the different segmentation problem instances. This points to a metaheuristic approach which could reuse the enhancements performed to a wider set of problems in a simpler way, specially bearing in mind the very different applications which these algorithms may have. Configuring a single objective optimization technique for segmentation optimization such as the one presented always has to deal with two different objectives: the cost of the technique and the quality of the results.

These initially suggests a multi-objective technique, particularly since quality measurements may also include different objective functions. Actually, for the required quality comparisons of the proposal, quality indicators extracted from multi-objective evolutionary algorithms literature were used (specifically, the hypervolume indicator).

Stopping criteria in evolutionary algorithms have suffered a certain lack of attention from the research community. There are many different reasons which have been analyzed: well-known datasets, budget focused research, stochastic nature of the algorithms, difficulties in convergence analysis... While whole algorithm proposals usually include specific choices for their different operators, such as selection, crossover, mutation or archiving, the stopping criteria is usually left out of them. This is a serious issue for the development of practical applications which have to deal with real problems with unknown complexity or required budget.

This handicap for the practical application of evolutionary algorithms has been covered both for the single-objective and multi-objective cases. Analyzing single-objective stopping criteria, this thesis's proposal started from an initial memetic algorithm including tracking of the search space. This algorithm lead to dynamic stopping criteria which presented a new approach: proposing stopping criteria which have an active role in the evolutionary cycle (preventing related problematic situations such as early convergence). The key to this active role is the presented diversity management.

The early results from the initial memetic algorithm lead to the extraction of its embedded artifacts to propose an independent stopping criterion which tracked both the variable and the objective space. The core of this proposal is based on two different artifacts: the gene matrix tracks the coverage of the search space by creating a matrix containing the different variables and the different subranges of their search space. Whenever an individual reached a certain subrange this change was updated in the matrix. This matrix has dynamically growing properties, such that a more thorough coverage of the search space is performed as required by the concrete problem instance. The gene matrix could be used to perform a passive stopping criteria, requiring a certain coverage before the algorithm is stopped.

The active artifact of the proposal is the mutagenesis operator. Mutagenesis is a guided mutation which modifies a certain gene value to cover a subrange which still has not been marked in the gene matrix. It is applied to the worst individuals in the population after selection has been performed, in order to lessen exploitation capabilities as little as possible. This mutation takes the active role of the stopping criteria: the loss of diversity is probably the most invoked reason for early convergence in evolutionary algorithm and, thus, instead of measuring this characteristic to stop the algorithm, the proposal is to enhance it to prevent the requirement to stop the algorithm. Since diversity is being actively modified, the stopping criterion still needs a characteristic according to which the stopping criterion can be triggered.

The stopping criterion is triggered according to a window of the best individual's function value, using a comparison based on a certain degree of improvement. The criterion was tested along with a canonical evolution strategy over a set of 27 optimization functions with different characteristics. The results show that the stopping proposal is capable of enhancing the exploratory capabilities of the algorithm, represented under two different typical results: if the exploratory capabilities of the underlying algorithm are not fit for the requirements of the problem (forced under low population sizes), the results are clearly improved. In a general application case, the results exhibited with the use of mutagenesis imply more robust stopping results, where the final results are considered satisfactory, effectively escaping those

local optima. The typical results under these circumstances are a lower average final function value with a higher median value. These results are caused by the additional cost which is spent in the mutagenesis procedure and the additional required function evaluations, which provide a slightly worse final results if the canonical technique is able, by its own exploratory capabilities, to reach a solution without falling into early convergence. Finally, if there is a clear path to the minimal solution, meaning that exploitation capabilities are the core of the evolutionary process, mutagenesis obtains worse results than its canonical alternative, since its exploration enhancements are not required for the given problem.

Facing single objective optimization provides with several interesting conclusions: the different tests required for the different statistical situations, the possible active role of the stopping process or the importance of diversity. For comparison purposes, if the results are not considered normal, literature typically uses a non-parametric test, such as Wilcoxon (not including the required skewness validation). However, for real validation of the results, other factors must be considered, such as the success of the algorithm in reaching final solutions or the mean value of the obtained results (since non-parametric tests are based on median). This makes these comparison processes harder than simple statistical comparisons to cover a complete view over the algorithm performance.

Multi-objective stopping criteria have been based on quality assessment developments, mainly the dimensionality reduction performed by quality indicators to a single value of the comparison of two Pareto fronts followed by the required processing. The proposal for this research line has been based on robustness, online nature of the procedures and simplicity. Initially, since different quality indicators are known to provide different characteristics regarding closeness to the real Pareto front and spread in the solution to their dimensionality reduction, a Kalman based fusion architecture was studied. The idea of this proposal is reusing traditional measures for quality assessment between different sets of Pareto fronts provided by different alternative algorithms and adapting them to an online comparison of different fronts obtained at consecutive generations of the running algorithm.

The issues regarding this proposal arise mainly from the lack of error information background regarding quality indicators and their effect. There are theoretical works regarding their applicability considering compatibility and completeness (which point to binary quality indicators as the only ones which can be used to really assess a comparison between different alternatives) but there is no quantitative error information regarding the different alternatives proposed (even though the complexity of the different indicators usually hints at their precision measuring these factors). For the linear estimation proposed, this creates the need to empirically determine the different error sources values (which are constant) and makes the use of traditional fusion architectures inapplicable, since further estimation of these error sources would only bias the final algorithm's results. A voting architecture has been used instead, where analysis of the different quality indicators is performed individually and each of them provide an assessment of the stopping generation, and according to these individual results, the final stopping decision is taken. The implemented procedure triggers the stopping generation when two out of the three quality indicators used have decided that the algorithm should be stopped. This mechanism, in spite of its simplicity, allows to enhance the robustness of the overall stopping criterion.

On the other hand, managing the two different errors, measurement (coming form the quality indicator use) and estimation process, allows the researchers to effectively control the point where additional improvements are considered unnecessary, bearing their computational

cost. This provides an additional degree of freedom for the configuration of the technique, which is not based on the point where no further statistically significant improvements are being obtained, but rather as a balance between quality and computational cost (a decision which is one of the foundations of stopping criteria). It is important to highlight that, once empirically determined, the values of these parameters have been tested over a set of different algorithms and test problems without further modifying their values, implying that there are problem and algorithm independent.

One of the main difficulties related to this testing is the lack of a best number of generations or function evaluations to compare the results to. Even though genetic drift is possible, it doesn't usually happen, and, thus, additional generations tend to provide slight improvements, such that stopping results always tend to be non-dominated (further computations cost implies better, or at least not worse, results).

The proposal for a multi-objective stopping criterion is based on the analysis of the issues detected during the Kalman voting architecture research and implementation: lack of proper error information which forces the empirical determination of constant values, complexity of available techniques which prevent their general application and other minor considerations. One of the clear objectives was to present a proposal which could be easy to configure and implement, in order to allow its application to different algorithms as a black box providing the required stopping criterion. This was also linked to the different errors which were present in the previous Kalman model about which no theoretical information could be found when applied to quality indicators. The result of both issues was to present a simple linear estimation based on least squares, which did not require any noise information.

The objective of the linear estimation is to determine when stagnation of the tracked quality indicator has occurred. To determine so, the normalized residue of the estimation is calculated (a similar residue had been used as part of the initial heuristic approach to segmentation in chapter 3). The residue follows a χ^2 distribution, which could be used to establish the appropriate thresholds to determine when the algorithm should be stopped. Following the implementation requirements to enhance its application, an additional approximation is performed, following the known mean and variance of the distribution and using Tchebycheff's inequality.

The result is a simple formula which does not require any statistical library and which introduces two requirements for the stopping triggering: a certain meaningfulness in the data and a certain value of improvement per generation. The data meaningfulness is required such that the linear representation has statistical representativeness, and is represented by threshold computed using the previously explained approximation of the χ^2 distribution of the residue. Once that has been stated, the user introduces a minimum amount of improvement in the quality indicator per generation, which is measured by the slope of the estimation.

Efficiency of the stopping criterion and its online behavior are the key considerations to the presented design, and have been considered at different steps. First of all, available approaches considered an analysis window which implied the recomputation of the different quality indicators in the window at every generation. This can be considered a semi-offline behavior (in fact, as covered in the fundamentals section, similar approaches have been taken in the segmentation domain to adapt offline algorithms to online data gathering and output requirements). The proposed approach considers, for every new generation, the computation of new quality indicator values only between the last and the previous generations (the binary quality indicators between these two Pareto fronts). After that, a window including the

previously computed values is considered, but does no require any additional calculations, enhancing the overall complexity of the criterion, particularly when it includes costly indicators such as hypervolume.

According to the iterative computation of the indicator values, also iterative computations of the different estimation parameters are proposed, providing an estimation which can be computed with a constant order complexity. This provides an efficient stopping criterion with a completely online behavior. The validation of the proposed technique has to face the same difficulty previously explained: the lack of proper quality measurements of a given stopping generation. A certain number of a-priori established generations are proposed for each problem (based on available values in the literature on this topic) and the Pareto fronts are calculated at the given stopping generation and the final one. After that, we provide information of the quality indicators between the preciously stated final Pareto fronts and the ones obtained with the stopping criterion, including stopping generation and hypervolume statistical values. These values again are not statistically meaningful due to the arbitrary final generations used, but provide insightful views over the applicability and robustness of the technique.

Least Squares Stopping Criterion (LSSC), the proposal described, shows that it provides robust results across the different algorithms and problems, and provides the researcher with an easily configurable set of parameters (all of which have suggested values). It is also easy to implement and does not require any additional libraries as some of the available techniques, highlighting its efficiency and online behavior, which cope with the presented objectives.

Finally, the evolutionary implementation for the segmentation domain is faced. This implementation includes most of the knowledge acquired in the development of the previously explained techniques, ranging from the different representations of the information in evolutionary computation to the requirements of dynamic stopping criteria, including other factors such as the importance of diversity. Initially, the multi-objective nature of the problem was studied. To do so, a thorough coverage of selected techniques from the domain is presented. These techniques are single-objective (heuristic and metaheuristic), and this analysis covers how they have coped with the underlying multi-objective nature of segmentation processes relating them to theoretical approximations. The conclusion is that they all require this multi-objective handling by different means (parametric, different orderings, aggregated functions...), leading to the a-posteriori proposal of a MOEA presented.

Once the MOEA approach has been selected, representation is considered. The key concept in a segmentation process is dominant points, the points in the original data which are the extremes of the final segments. The idea of a multi-objective approach is that, as shown in the presented analysis, the position of key dominant points is not heavily altered at different compression levels (the different individuals of the Pareto front) and so they can share valuable information during the evolutionary process. Two possible representations are considered: a genetic algorithm stating with a 1 or a 0 whether the point is considered dominant or not or an evolutionary algorithm which includes only the numbering of dominant points (and these number would be repeated). This last approach enhances the preservation of those key dominant points previously commented, but increases the size of the search space. The genetic approach elitism was considered sufficient for the preservation of the important dominant points, particularly since the multi-objective proposal spreads them through the Pareto front, and was chosen for this task.

The chosen algorithm was SPEA2, due to its archive technique and extensive use in

the research community. It is interesting to consider that the final objective of the MOEA in a segmentation problem is to obtain one individual for each possible compression level, establishing the value of the archive size according to the problem instance size. Finally, initialization issues are faced. Traditional initialization procedures look for diversity in the variable space. An alternative initialization is included based on a memetic approach using two complimentary local search procedures: bottom-up and top-down. This initialization has to deal with the single-objective nature of the presented techniques, such that different individuals of the Pareto front have to be obtained using different runs with different configurations of the algorithm, and obtaining well spread Pareto fronts of initialization individuals can be difficult.

One additional initialization procedure is considered based on the problem's characteristics. Diversity in the objective space is generally considered an important feature of MOEAs, such that an initialization looking for this characteristic is proposed, named uniform initialization. This underlying idea is that the two considered objectives of the problem, number of dominant points and representation error, are heavily in conflict, such that providing a good spread in one of them is bound to provide a good spread in the other (and, thus, in the generated Pareto front). A random number of dominant points is generated for each individual in the Pareto front, and, afterwards, these dominant points are randomly placed in the original data. The result of this technique are Pareto fronts that have a much better diversity in objective space than any of the other two alternatives, without requiring representative additional computational costs to the variable space diversity proposal.

The validation is performed over a set of three traditional figures from the polygonal approximation domain, in order to be able to compare the obtained results versus a complete set of alternatives. Eight specific heuristic approaches are included, and also a single-objective evolutionary approach. The comparisons are based on the appropriate statistical testing performed over thirty runs of the proposed MOEA, comparing the obtained solution with the same degree of compression to its heuristic alternative. In the evolutionary approach case, several runs provided several individuals, such that individual comparisons and whole Pareto fronts were compared. Initially, the initialization alternatives are compared among each other. Uniform initialization yields better final results at a lower cost than the local search based approach, leading to its choice as the initialization technique for the final algorithm results.

The achieved results have a growing number of generations for the three problem instances, where the quality of the obtained results is compared among the different results, until for one the figures, the increase in the number of generations provides no further statistically significant improvements. The same configuration is applied to the three figures. Comparing the results to their heuristic alternatives, the achieved results were significantly better in 21 out of 24 cases, being worse only in one case. The comparison with the evolutionary alternative was significantly better in 15 out of 16 cases, and worse in 1. Also, comparing the obtained Pareto fronts (restricting the number of individuals obtained in the MOEA approach, to obtain unbiased results) shown statistically better results in terms of hypervolume value towards the proposed approach.

The general MOEA approach to segmentation took benefit from its multi-objective nature, but there were still opened questions. The algorithm configuration had been extracted from a-posteriori results, which were valuable for the comparison versus alternative techniques but didn't provide with clear values for new problems. The complexity of the environmental selection in the archiving technique was also an issue, since it hampered the scalability of the

technique. Finally, it hadn't included an stopping criterion.

The final proposal considers each of the evolutionary individual steps and adapts them to the segmentation domain. The chromosome adds additional information to allow partial recomputations of the fitness values, performed by fitness-aware operators, which calculate children fitness values according to their parents ones and the changes performed. A novel archiving procedure is also introduced, to cope with the excessive complexity of the environmental selection performed in SPEA2. This archiving procedure uses the bi-objective nature of the problem, with one of them discrete, to provide a relaxed version of the dominance concept. This novel archiving technique provides a scalable procedure which also allows parallelization opportunities.

Local search techniques are also reviewed presenting multi-objective versions of their basic proposals, using the modified chromosome representation introduced. These versions can effectively obtain Pareto fronts of solutions to be included as initialization processes. Even though these initial fronts are better (in terms of quality assessment) than the ones obtained with the uniform technique, these improvements are not reflected in the final hypervolume results. This seems to be caused by the biased search which the local search techniques produce, which leads to faster good results but hamper the exploration capabilities of the algorithm. For this reason they have not been included in the final algorithm proposal, even though they cannot be discarded, and their use is recommended in order to obtain a satisfactory solution faster. In any case, they are extremely useful to configure the archive size, since they find zero error individuals with a lower value than the problem instance dimensionality, if they exist, and thus allow a better search focus.

Finally, the stopping criteria designed have also been adapted to the segmentation domain, introducing similar considerations to the archiving technique. Since the archive is capable of holding the entire Pareto front, the comparisons for the stopping criterion are simplified. A boolean progress indicator has been designed which simply determines whether there has been any progress at any compression level during the analyzed generation (in fact, in the proposed algorithm, there are not generations properly speaking, but rather boundaries for the number of transformation operations before the progress is checked). If there has been no progress at any compression level for a certain window of generations, the algorithm is stopped. This technique shows satisfactory stopping generations and only needs one configuration parameter: the window size. Also, the used progress indicator can be computed seamlessly as part of the archiving procedure

This final proposal integrates the knowledge acquired through the different previous chapters, if not the devised procedures themselves. It is scalable, provides a solution for each compression level, has an integrated stopping criterion which can be easily configured and is competitive with the single objective approaches available in the literature. It also allows the DM to consider which segmentation solution suits him better once the solutions have been computed, instead of trying to configure the technique a-priori to guide the technique solution to the one presumably desired, simplifying the configuration process.

8.2 Future lines

Three main topics with strong relationships among them have been covered in this thesis: single-objective stopping criteria, multi-objective stopping criteria and multi-objective evolutionary approaches to the segmentation issue. The core proposal for single-objective

optimization has been the reconsideration of the passive stopping criterion role, changing it towards an active one based on diversity. Future lines can be based on the different diversity measures available on the literature, studying their behavior under different guided mutation operators in order to avoid early convergence. Possible alternatives to be studied are the correlations between different guided mutation operators (such as including an exploration enhancement over an exploitation guided algorithm like CMA-ES), diversity guided alternatives providing self-stopping capabilities or multi-objective approaches were different measures of diversity could be considered as an objective and let the algorithm guide the evolution accordingly. The measurement of diversity itself is an open issue, heavily impacting any future algorithm trying to modify its value. From the available results, the operator proposed is useful at certain problem configurations and different moments during the evolutionary approach, leading to a possible dynamic self-configuration of the algorithm, where run-time measures of performance could lead to the application of the guided mutation towards a more focused exploration or exploitation behavior.

According to the established statistical comparison procedures, which rely on non-parametric statistical testing when the distributions are not considered normal (something which happens frequently in evolutionary results) it would be interesting to consider additional or alternative approaches. For instance, at the cost of additional exploration computational cost, a technique may exhibit results where no early convergence has been detected in an algorithm (obtaining a 100% success rate). Compared to different alternatives more focused on exploitation where those early convergence cases happen to a certain extent, these techniques tend to exhibit a higher median value with a lower mean value on the final function results. If non-parametric tests are run without considering the skewness of the results, statistically better results will favor the second technique, providing an incomplete assessment for the algorithm choice.

In multi-objective stopping criteria, the main focus covered in the literature is based on the comparison of numeric values coming the dimensionality reduction performed by quality indicators. However, proposals based on whole Pareto front analysis remain to be proposed, following the quality assessment proposals available in the literature. The main pitfall to be covered in this case is the lack of a proper comparison process, since no optimal stopping generation can be stated, leading to ad-hoc comparison methods which difficult the introduction of these techniques in algorithm proposals. Combining stopping criteria into the evolutionary cycle of indicator based approaches, which leads to the reuse of the results from the indicators used by the underlying algorithm, would provide criteria with an insignificant computational cost impact, requiring a joint choice of guiding indicators which could be used by both the algorithm and the embedded stopping criterion. Another issue to be faced is the proposal of iterative normalization procedures which can be integrated into online stopping criteria such as the proposal included in this work, unlike the window based approaches where the re-normalization implies the recomputation of quality indicators in the whole window of Pareto fronts.

The establishment of sound and robust stopping criterion opens a new dimension to quality comparison, since traditional approaches were based on a certain predefined computation budget. Instead of comparing only the quality of the final obtained Pareto fronts, quality comparison must be faced as a multi-objective problem, regarding the quality of the obtained results and the cost of these results. This approach to quality assessment would provide a less biased result towards more complex algorithms, which actually incur in a higher

computational cost without any impact in their quality comparisons.

The proposal presented for the segmentation issue has proved to be competitive in terms of quality of the final results. The initially considered memetic approach was discarded at the initialization step, but could be reintegrated as part of the mutation, since representative improvements in one individual are propagated to the whole Pareto front. Different MOEAs can also be considered as the basic algorithm, comparing their different performances. The uniform initialization proposed could be extended to different algorithms, handling the additional computational costs implied with the final benefits in the resultant Pareto fronts, considering the application of general local search techniques for this process. Since the metaheuristic proposal has one of its foundations in the adaptation to different domains, such adaptations should be researched, considering noisy domains, possible online applications, multiple model segmentation or the inclusion of domain based restrictions.

Considering the modifications performed over the general MOEA for the final proposal, different future lines are opened regarding the different improvements introduced. The presented alternative representation opens the study of the applicability of proposed partial fitness codification and fitness aware operators to a wider set of problems and domains. The alternative archiving technique can be combined at certain steps with full Pareto-dominance approaches, or other mixed approaches where more than one individual can be introduced for each compression level, in order to increase the diversity. Regarding initialization and the use of local search approaches, future lines should research the avoidance of the diversity loss introduced by them (probably combined with an enhanced archiving). Also, the presented multi-objective local search algorithms allow the introduction of new hyperheuristic or memetic approaches combining them and evolutionary algorithms.

A

Appendix: Single Optimization Function Set

> *The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents We live on a placid island of ignorance in the midst of black seas of infinity, and it was not meant that we should voyage far The sciences, each straining in its own direction, have hitherto harmed us little; but some day the piecing together of dissociated knowledge will open up such terrifying vistas of reality, and of our frightful position therein, that we shall either go mad from the revelation or flee from the deadly light into the peace and safety of a new dark age*
>
> Howard Phillips Lovecraft, *The Call of Cthulhu*, 1926

In chapter 4 an approach to single objective stopping criterion was presented, along with its inclusion into a memetic algorithm, whose performance was compared with CMAES algorithm (section 4.2.3). To perform this comparison, a set of twenty-seven functions was chosen according to their set of characteristics. This set was overviewed in table 4.2. This chapter presents the complete description of this set.

A.1 Function set complete description

This additional section includes the formulation, dimensionality, search space and bidimensional representation of all the different functions included in this chapter's results dataset.

1. *Ackley function*. Figure A.1

   dimensionality: n (30)

   Search space: $-32 \leq x_i \leq 32$

$$f_1(x) = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)} \tag{A.1}$$

2. *Beale function*. Figure A.2

   dimensionality: 2

Search space: $-4.5 \leq x_i \leq 4.5$

$$f_2(x) = (1.5 + x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \quad \text{(A.2)}$$

3. *Bohachevsky function*. Figure A.3

   dimensionality: 2

   Search space: $-100 \leq x_i \leq 100$

$$f_3(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \quad \text{(A.3)}$$

4. *Booth function*. Figure A.4

   dimensionality: 2

   Search space: $-10 \leq x_i \leq 10$

$$f_4(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad \text{(A.4)}$$

5. *Branin function*. Figure A.5

   dimensionality: 2

   Search space: $-5 \leq x_1 \leq 10$, $0 \leq x_2 \leq 15$

$$f_5(x) = (x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}cos(x_1) + 10 \quad \text{(A.5)}$$

6. *Colville function*. Figure A.6

   dimensionality: 4

   Search space: $-10 \leq x_i \leq 10$

$$\begin{aligned} f_6(x) =& 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 \\ &+ 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_1 - 1)(x_3 - 1) \end{aligned} \quad \text{(A.6)}$$

7. *Dixon and Price function*. Figure A.7

   dimensionality: n (30)

   Search space: $-10 \leq x_i \leq 10$

$$f_7(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2 \quad \text{(A.7)}$$

8. *Easom function*. Figure A.8

   dimensionality: 2

   Search space: $-100 \leq x_i \leq 100$

$$f_8(x) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2} \quad \text{(A.8)}$$

9. *Goldstein and Price function*. Figure A.9

   dimensionality: 2

   Search space: $-2 \leq x_i \leq 2$

$$f_9(x) = \left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right) * \left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 - 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)\right) \tag{A.9}$$

10. *Griewank function*. Figure A.10

    dimensionality: n (30)

    Search space: $-600 \leq x_i \leq 600$

$$f_{10}(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{A.10}$$

11. *Hartmann function*. Figure A.11

    dimensionality: 6

    Search space: $-600 \leq x_i \leq 600$

$$\begin{aligned} f_{11}(x) &= -\sum_{i=1}^{4} \alpha_i e^{\sum_{j=1}^{6} B_{ij}(x_j - Q_{ij})^2} \\ \alpha &= \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix} \quad B = \begin{bmatrix} 10 & 3 & 17 & 3.05 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix} \\ Q &= 10^{-4} \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix} \end{aligned} \tag{A.11}$$

12. *Hump function*. Figure A.12

    dimensionality: 2

    Search space: $-5 \leq x_i \leq 5$

$$f_{12}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{A.12}$$

13. *Levy function*. Figure A.13

    dimensionality: n (30)

    Search space: $-10 \leq x_i \leq 10$

$$\begin{aligned} f_{13}(x) = &\sin^2(\pi y_1) + \sum_{i=1}^{n} -1(y_i - 1)^2 \left(1 + 10\sin^2(\pi y_i + 1)\right) + \\ &+ (y_n - 1)^2 \left(1 + 10\sin^2(2\pi y_n)\right) \qquad y_i = 1 + \frac{x_i - 1}{4} \end{aligned} \tag{A.13}$$

14. *Matyas function*. Figure A.14

    dimensionality: 2

    Search space: $-10 \leq x_i \leq 10$

$$f_{14}(x) = 0.26(x_0^2 + x_1^2) - 0.48x_0x_1 \tag{A.14}$$

15. *Michalewics function*. Figure A.15

    dimensionality: 10

    Search space: $0 \leq x_i \leq \pi$

$$f_{15}(x) = -\sum_{i=1}^{n} \sin(x_i) \left(\sin\left(\frac{ix_i^2}{\pi} \right) \right)^{2m} \qquad m = 10 \tag{A.15}$$

16. *Perm function*. Figure A.16

    dimensionality: n (30)

    Search space: $-n \leq x_i \leq n$

$$f_{16}(x) = \sum_{k=1}^{n} \left(\sum_{i=1}^{n} (i + \beta) \left((x_i)^k - (i)^{-k} \right) \right)^2 \qquad \beta = 0.5 \tag{A.16}$$

17. *Powell function*. Figure A.17

    dimensionality: 28

    Search space: $-4 \leq x_i \leq 5$

$$f_{17}(x) = \sum_{i=1}^{n/4} (x_{4j-4} + 10x_{4j-3})^2 + 5(x_{4j-2} - x_{4j-1})^2 + (x_{4j-3} - 2x_{4j-2})^4 + \\ + 10(x_{4j-4} - x_{4j-1})^4 \tag{A.17}$$

18. *Power-sum function*. Figure A.18

    dimensionality: n (30)

    Search space: $0 \leq x_i \leq n$

$$f_{18}(x) = \sum_{k=1}^{n} \left(\left(\sum_{i=1}^{n} x_i^k \right) - b_k \right)^2 \tag{A.18}$$

19. *Rastrigin function*. Figure A.19

    dimensionality: n (30)

    Search space: $-5.12 \leq x_i \leq 5.12$

$$f_{19}(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right) \tag{A.19}$$

20. *Rosenbrock function*. Figure A.20

    dimensionality: n (30)

    Search space: $-5 \leq x_i \leq 10$

$$f_{20}(x) = \sum_{i=1}^{n-1} \left(100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2\right) \tag{A.20}$$

21. *Schwefel function*. Figure A.21

    dimensionality: n (30)

    Search space: $-500 \leq x_i \leq 500$

$$f_{21}(x) = 418.9829n - \sum_{i=1}^{n} \left(x_i \sin\left(\sqrt{x_i}\right)\right) \tag{A.21}$$

22. *Shekel function*. Figure A.22

    dimensionality: 4

    Search space: $0 \leq x_i \leq 10$

$$\begin{aligned} f_{22}(x) &= -\sum_{j=1}^{m} \left(\sum_{i=1}^{4} \left(x_i - C_{ij}\right)^2 + \beta_j\right)^{-1} \qquad m = 10 \\ \beta &= \frac{1}{10}\begin{bmatrix} 1 & 2 & 2 & 4 & 4 & 6 & 3 & 7 & 5 & 5 \end{bmatrix} \\ C &= \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix} \end{aligned} \tag{A.22}$$

23. *Shubert function*. Figure A.23

    dimensionality: 2

    Search space: $-10 \leq x_i \leq 10$

$$f_{23}(x) = \left(\sum_{i=1}^{5} i \cos\left((i+1)x_1 + i\right)\right)\left(\sum_{i=1}^{5} i \cos\left((i+1)x_2 + i\right)\right) \tag{A.23}$$

24. *Sphere function*. Figure A.24

    dimensionality: n (30)

    Search space: $-5.12 \leq x_i \leq 5.12$

$$f_{24}(x) = \sum_{i=1}^{n} x_i^2 \tag{A.24}$$

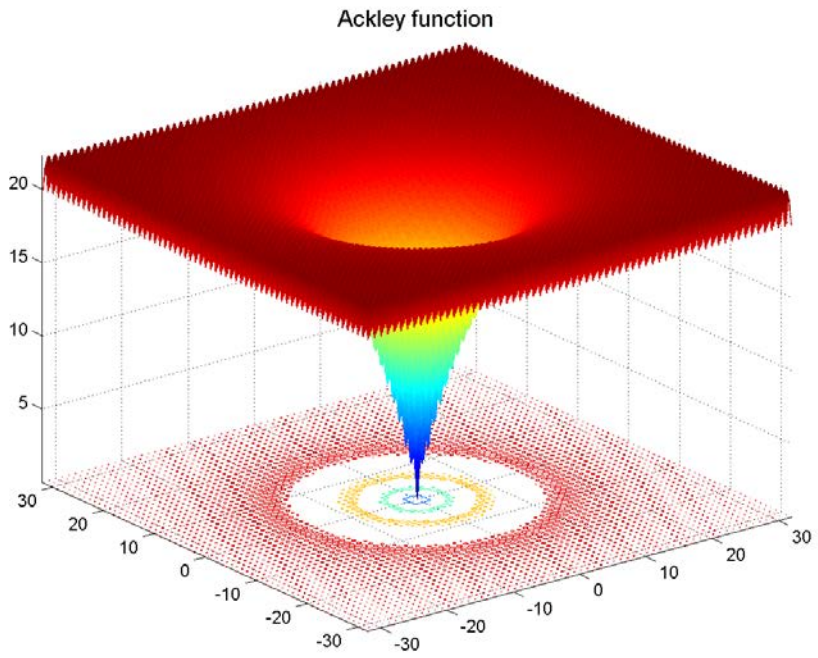


Figure A.1: Ackley function

25. *Sum Squares function*. Figure A.25
dimensionality: n (30)
Search space: $-10 \leq x_i \leq 10$

$$f_{25}(x) = \sum_{i=1}^{n} i x_i^2 \tag{A.25}$$

26. *Trid function*. Figure A.26
dimensionality: 10
Search space: $-n^2 \leq x_i \leq n^2$

$$f_{26}(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1} \tag{A.26}$$

27. *Zakharov function*. Figure A.27
dimensionality: n (30)
Search space: $-5 \leq x_i \leq 10$

$$f_{27}(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i \right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i \right)^4 \tag{A.27}$$

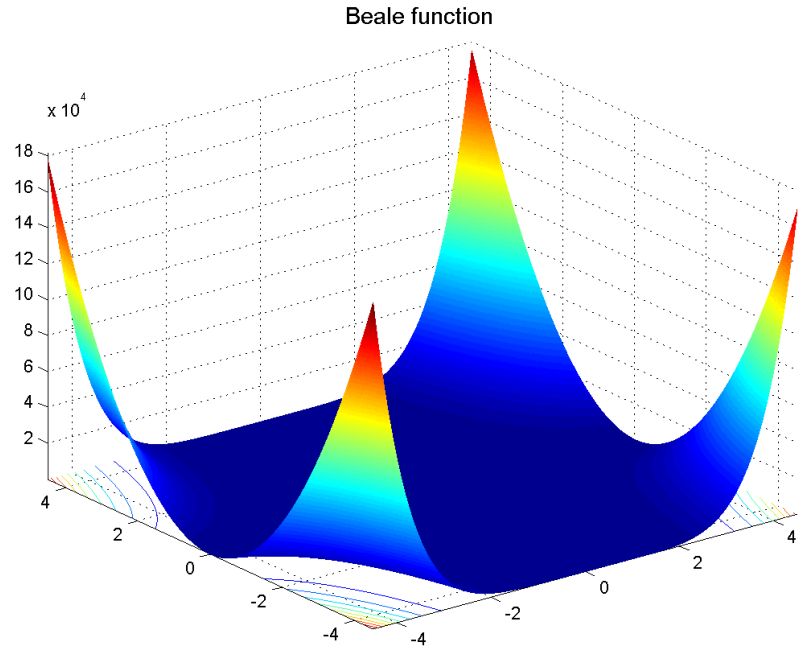


Figure A.2: Beale function

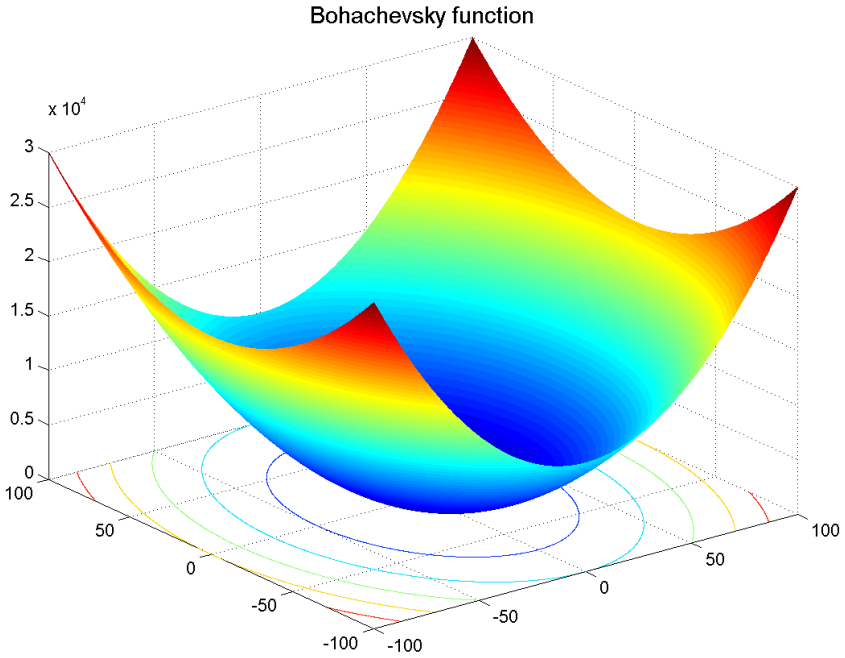


Figure A.3: Bohachevsky function

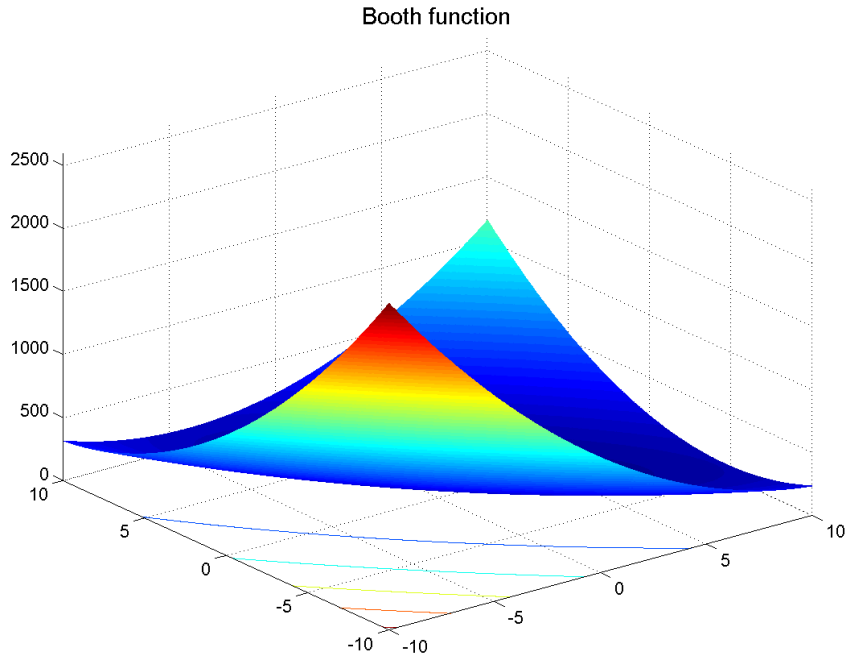


Figure A.4: Booth function

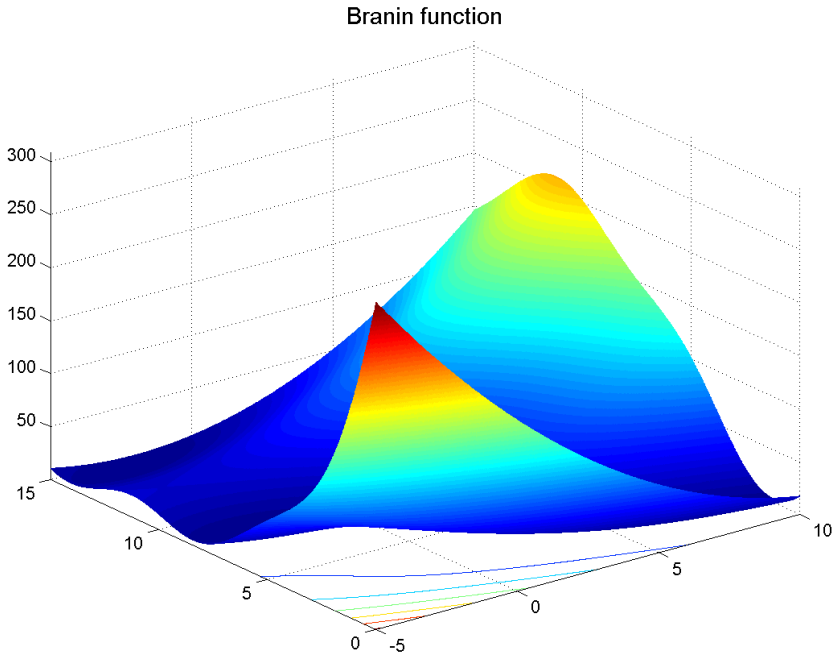


Figure A.5: Branin function

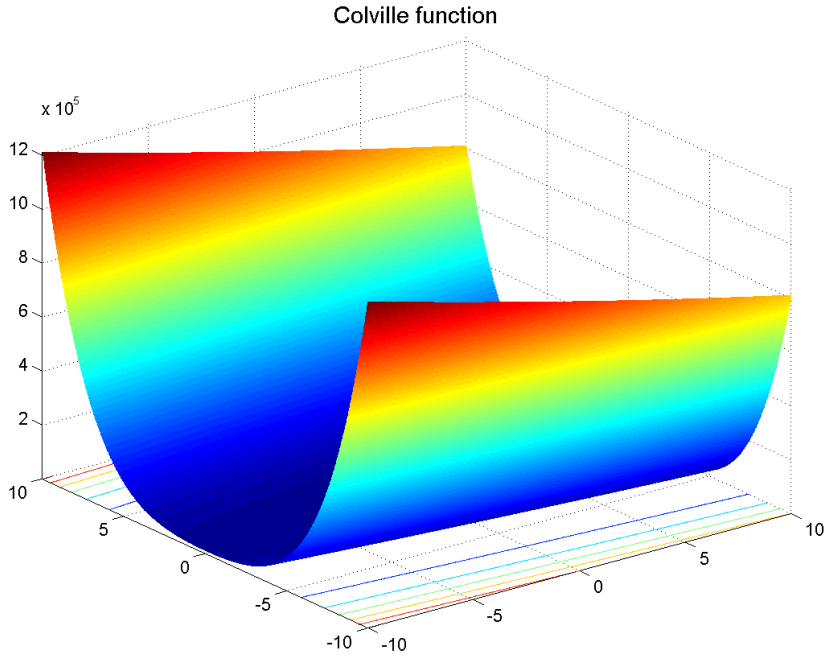


Figure A.6: Colville function

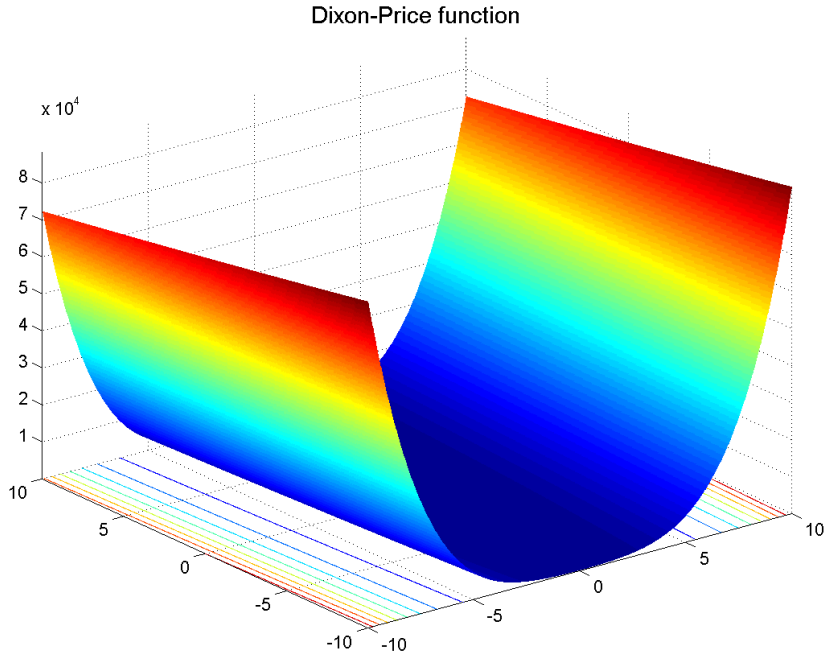


Figure A.7: Dixon and Price function

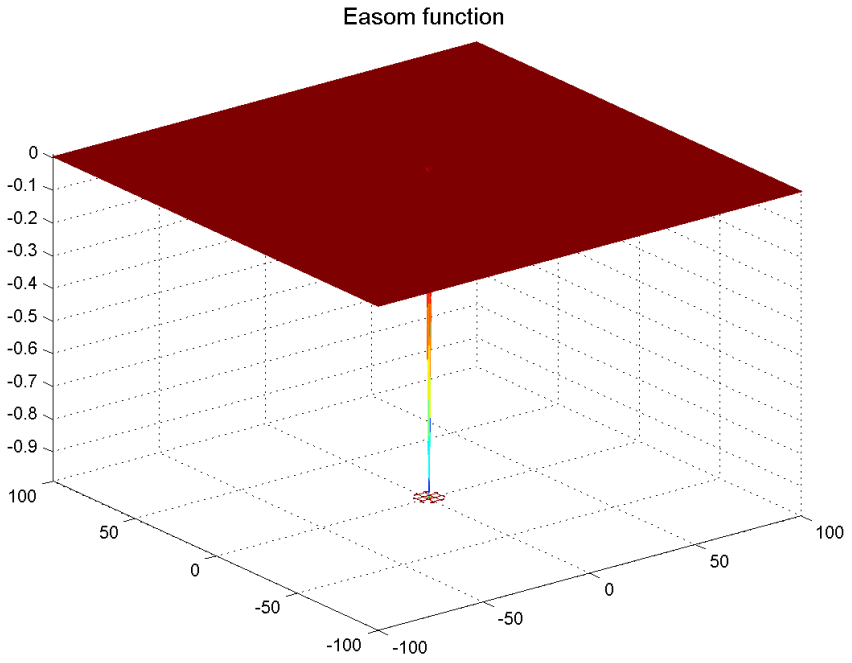


Figure A.8: Easom function

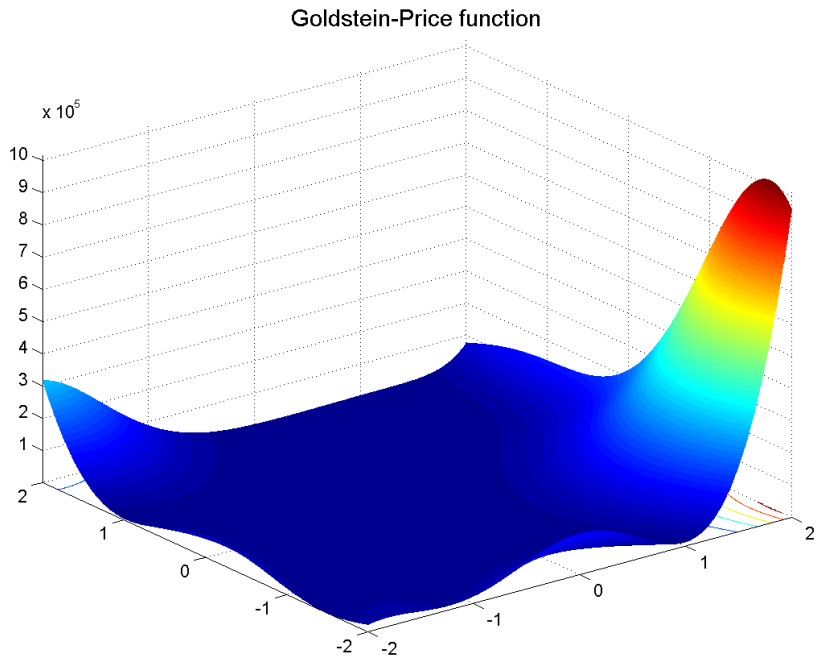


Figure A.9: Goldstein and Price function

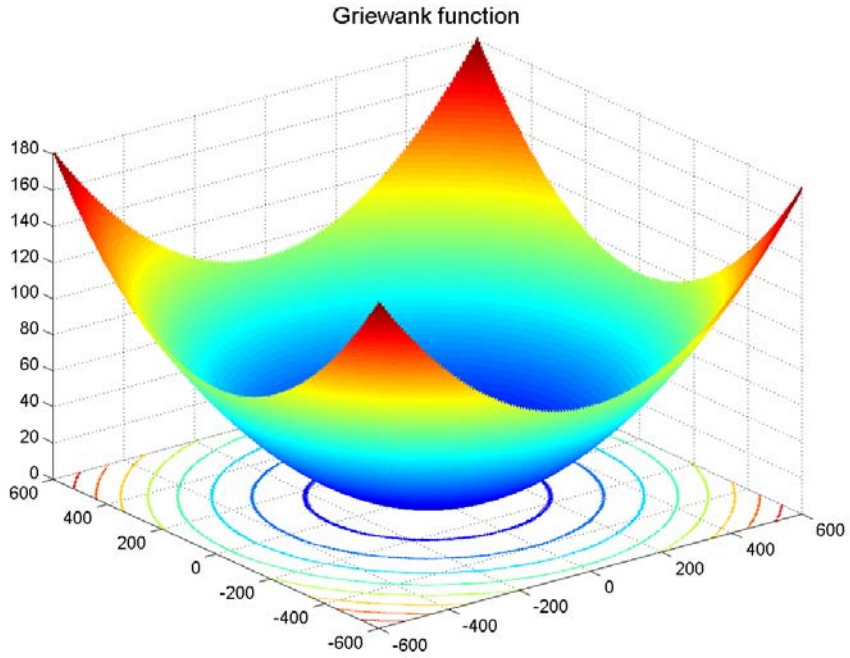


Figure A.10: Griewank function

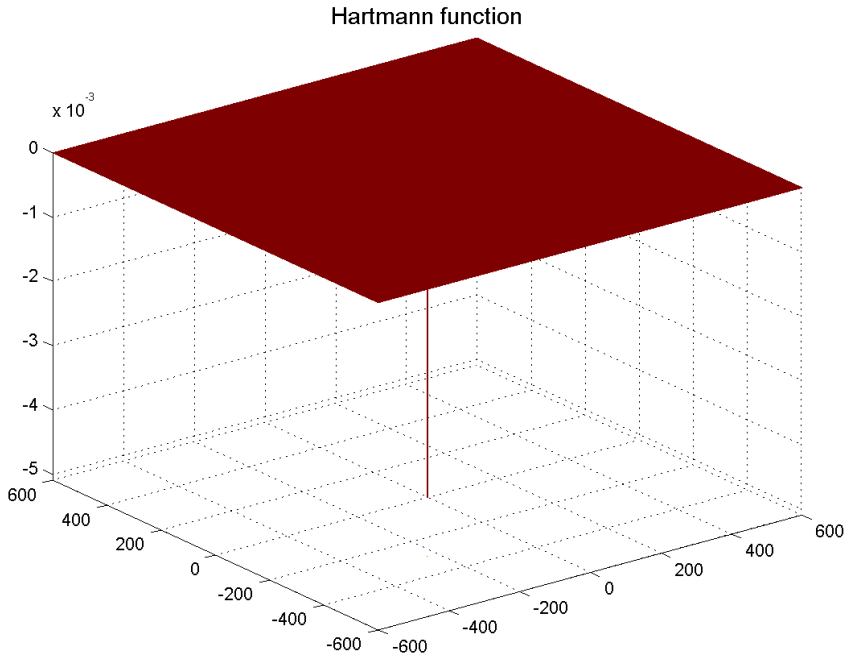


Figure A.11: Hartmann function

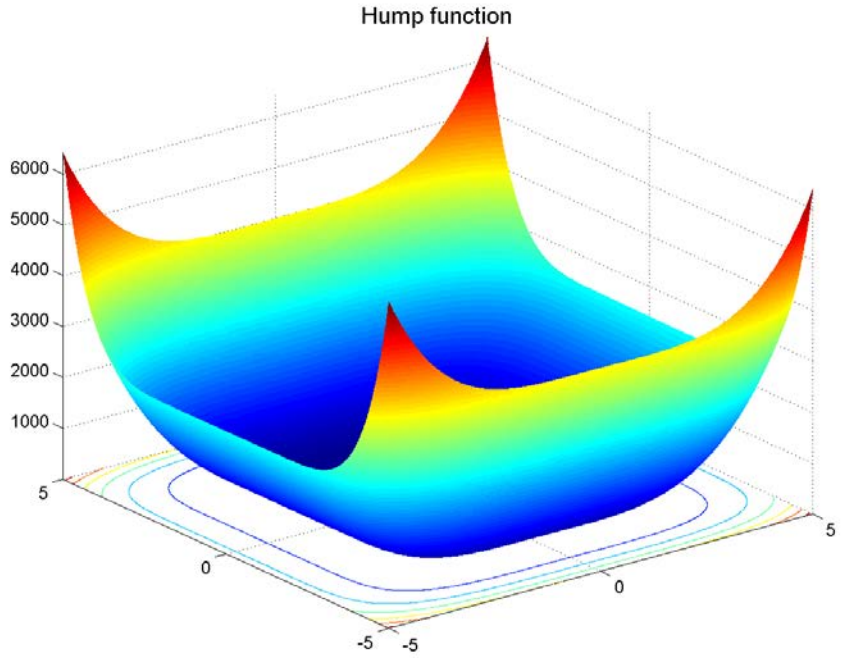


Figure A.12: Hump function

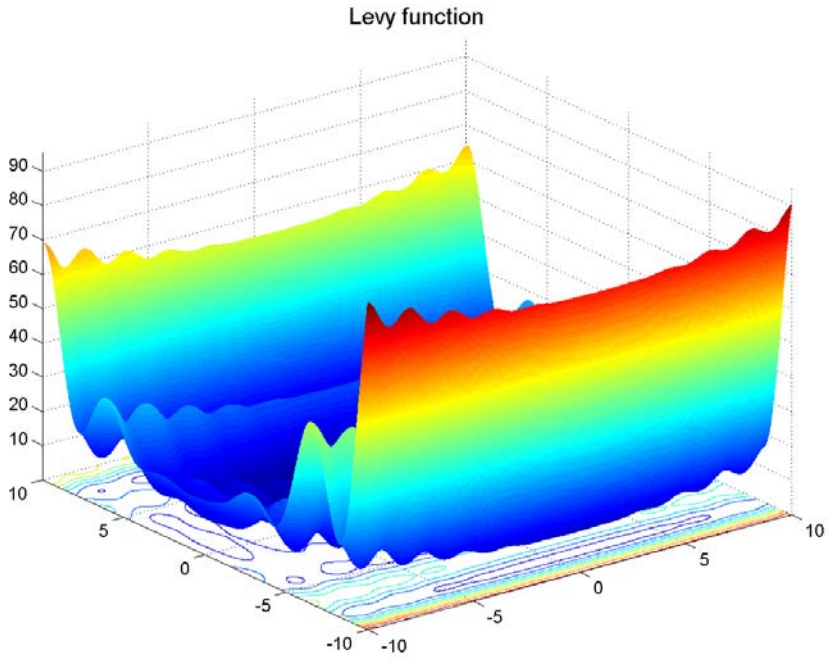


Figure A.13: Levy function

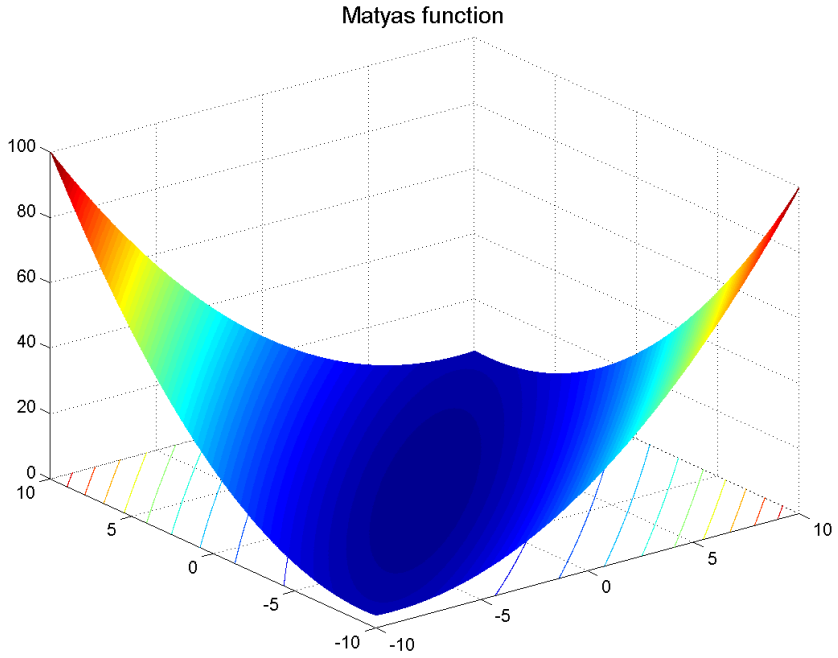


Figure A.14: Matyas function

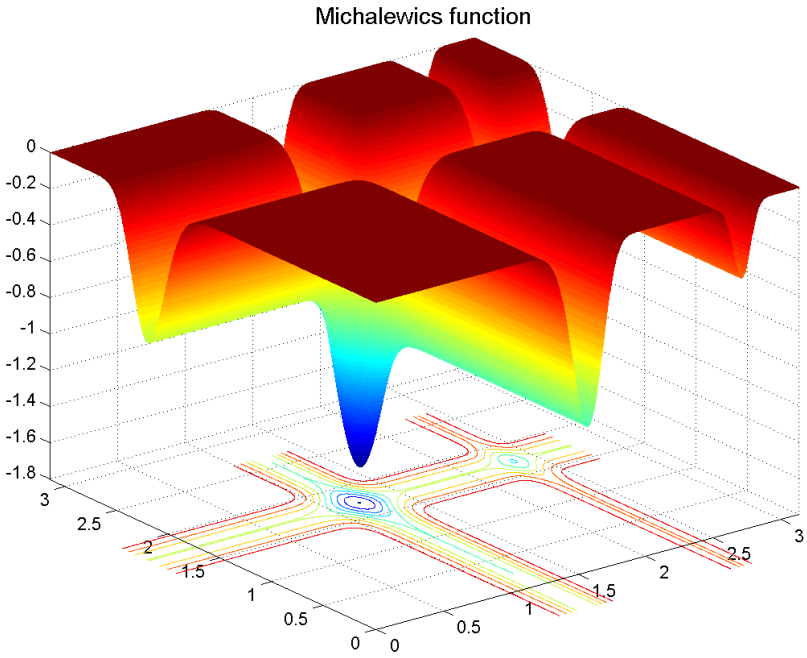


Figure A.15: Michalewics function

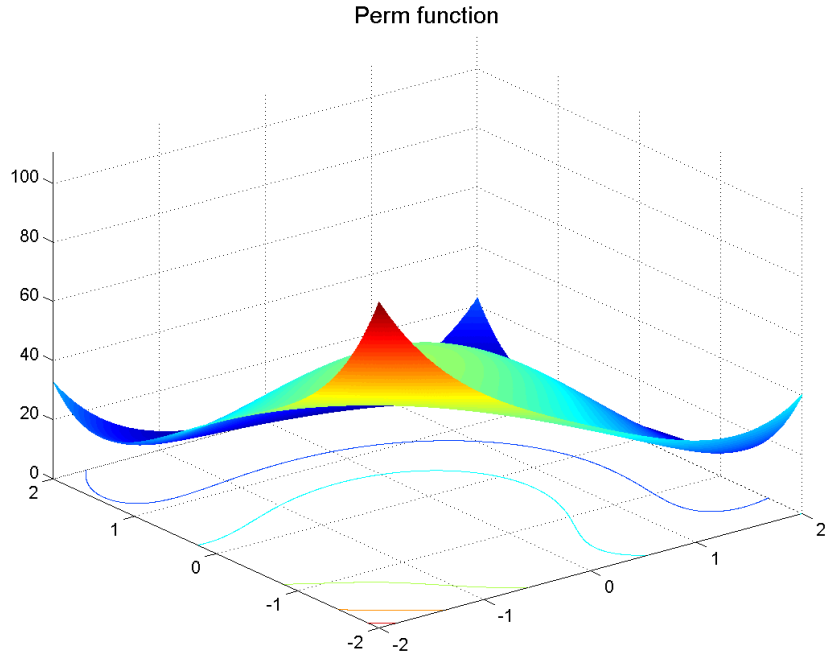


Figure A.16: Perm function

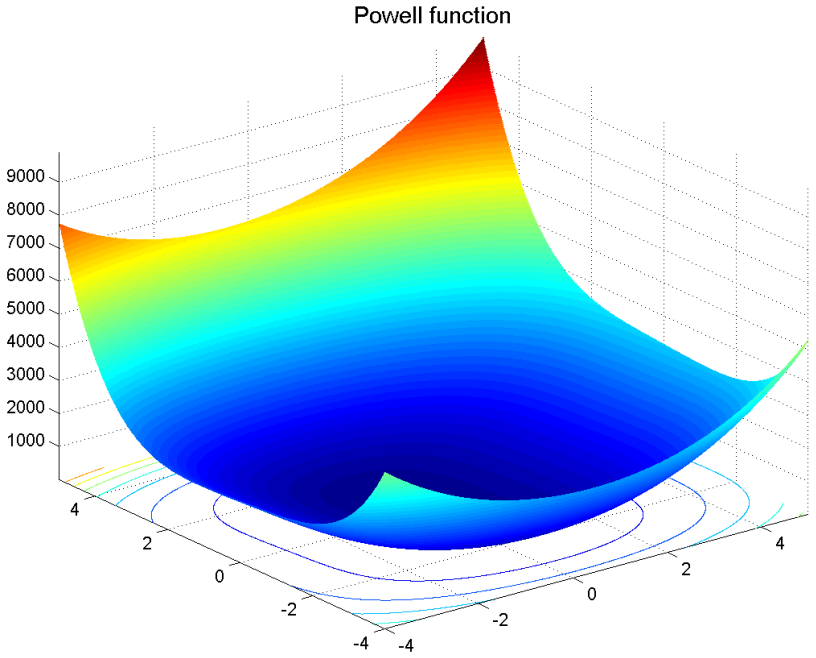


Figure A.17: Powell function

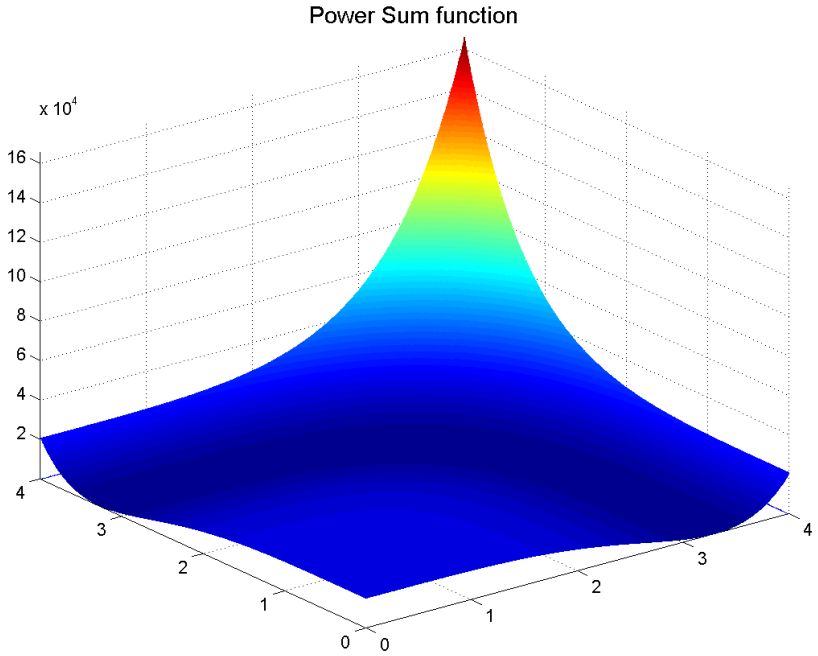


Figure A.18: Power Sum function

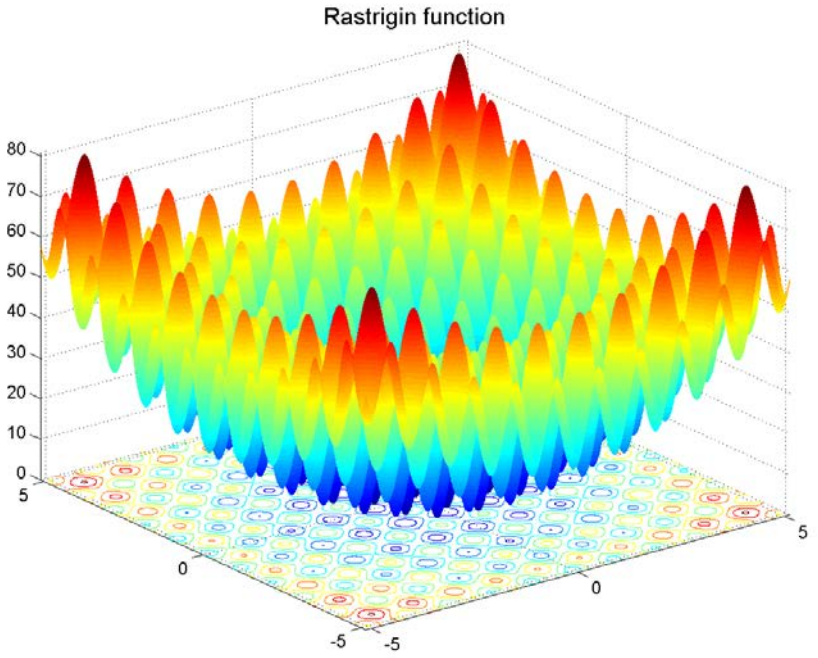


Figure A.19: Rastrigin function

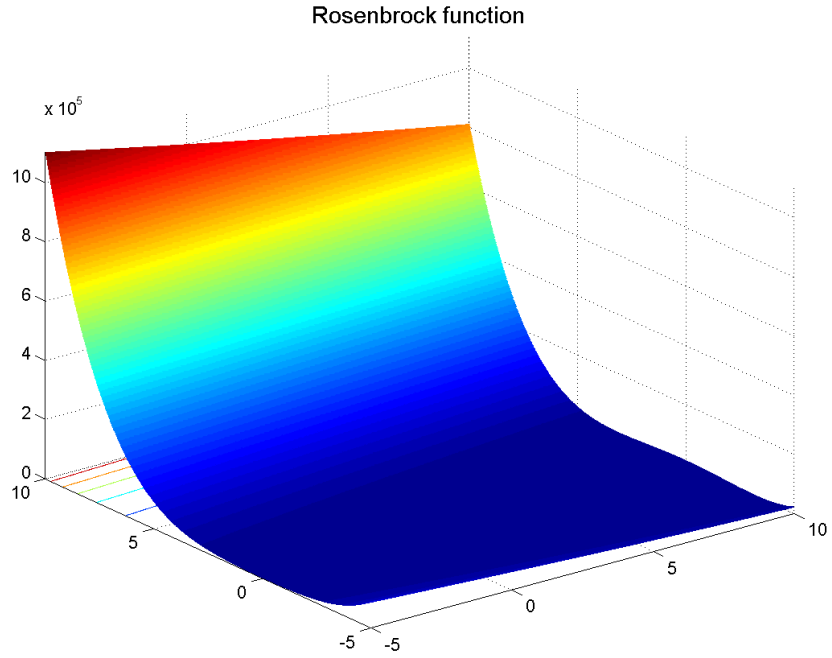


Figure A.20: Rosenbrock function

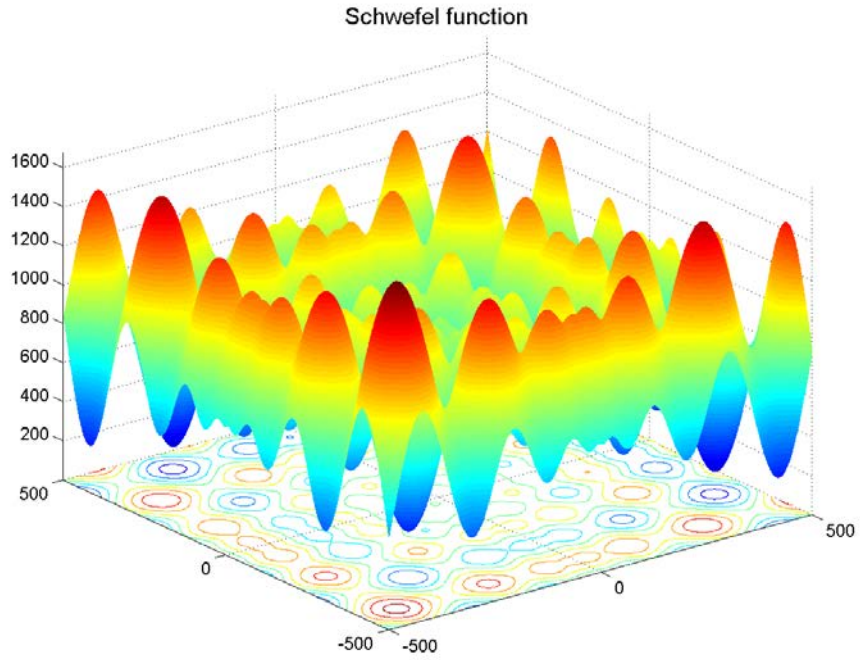


Figure A.21: Schwefel function

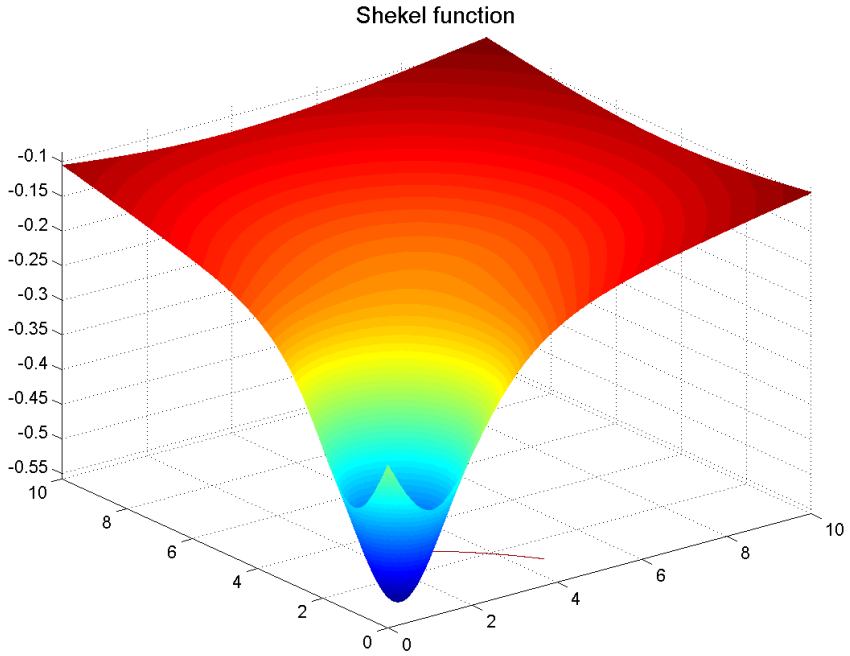


Figure A.22: Shekel function

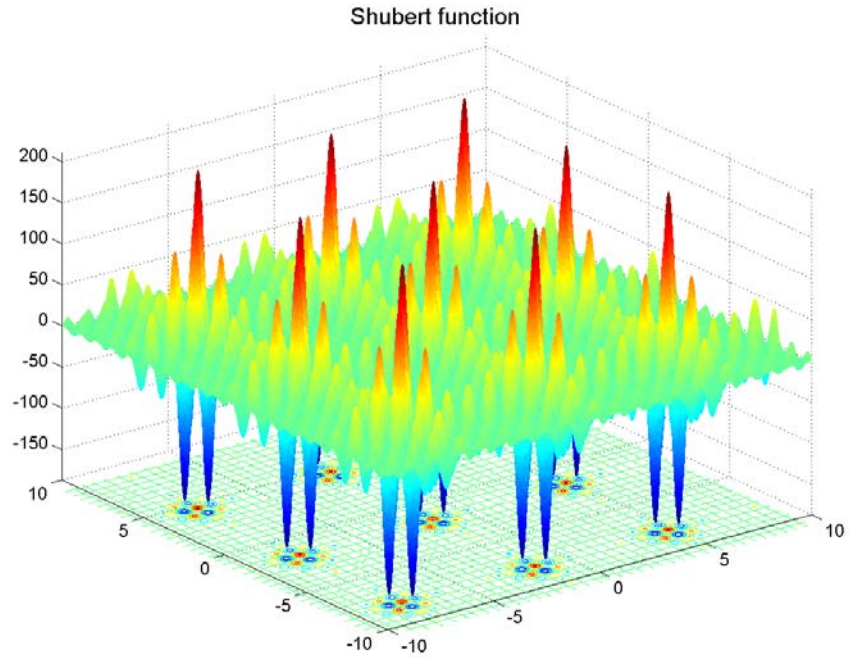


Figure A.23: Shubert function

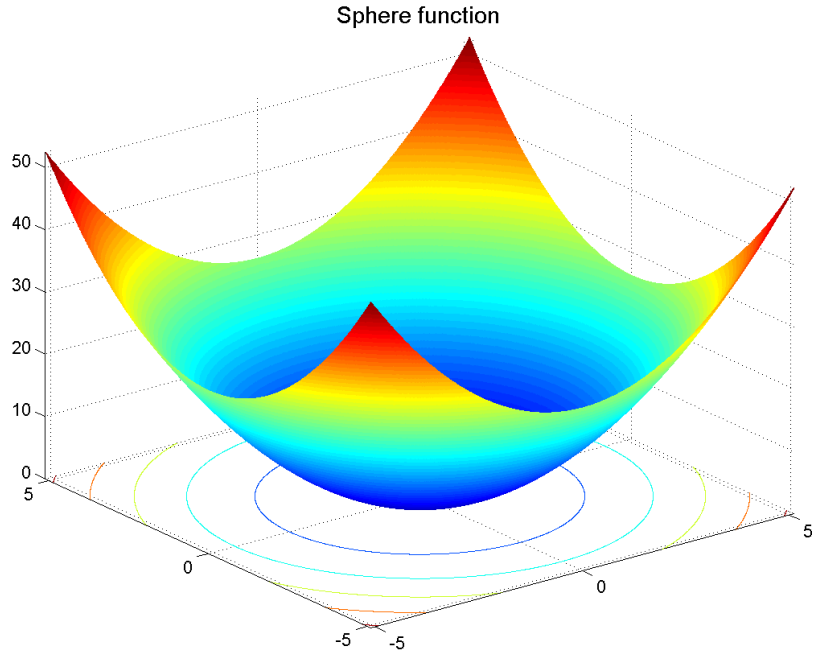


Figure A.24: Sphere function

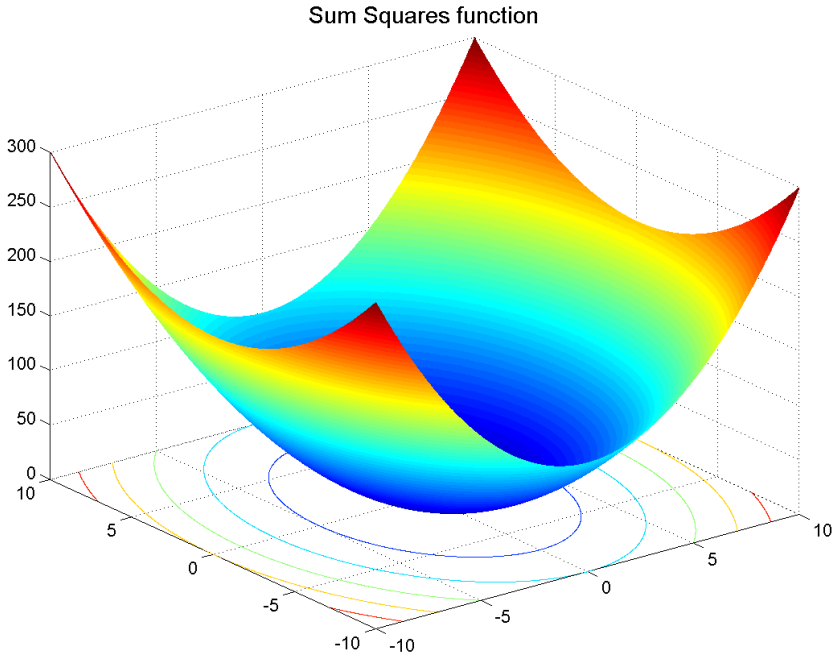


Figure A.25: Sum Squares function

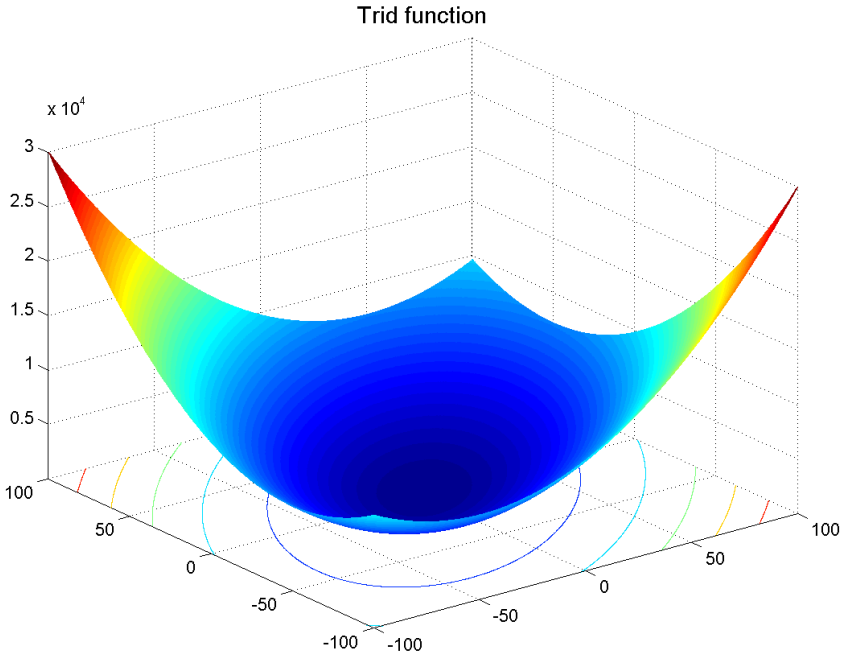


Figure A.26: Trid function

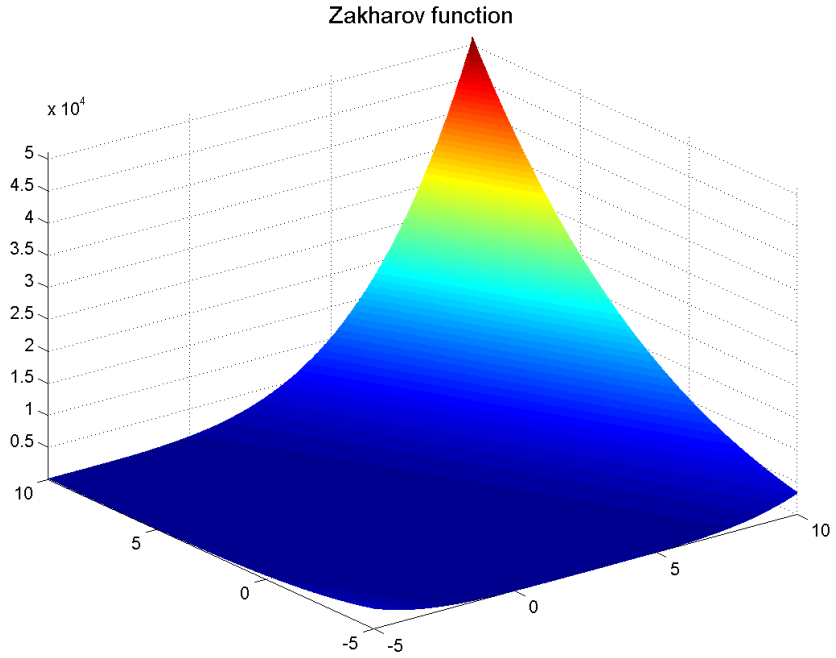


Figure A.27: Zakharov function

List of author's publications

JCR journal publications

Guerrero, J. L., Berlanga, A., & Molina, J. M. (2012). A multi-objective approach for the segmentation issue. *Engineering Optimization*, *44*(3), 267–287.

Guerrero, J. L., García, J., & Molina, J. (2011). Piecewise linear representation segmentation in noisy domains with a large number of measurements: the air traffic control domain. *International Journal on Artificial Intelligence Tools*, *20*(2), 367–399.

Journal publications

Guerrero, J. L., Berlanga, A., & Molina, J. M. (2014a). A guided mutation operator for dynamic diversity enhancement in evolutionary strategies. *International Journal of Natural Computing Research*, *4*(2), 20–39.

Guerrero, J. L., Berlanga, A., & Molina, J. M. (2014b). Multiobjective local search as an initialization procedure for evolutionary approaches to polygonal approximation. *International Journal of Artificial Intelligence*, *12*(1), 150–165.

Book chapter publications

Guerrero, J. L., García, J., & Molina, J. (2010). Air traffic trajectories segmentation based on time-series sensor data. In *Sensor Fusion and its Applications*, (pp. 31–52). Intech.

Congress publications

García, J., Guerrero, J. L., Luis Bustamante, Á., & Molina, J. M. (2010). Robust sensor fusion in real maritime surveillance scenarios. In *Proceedings of the 13th Conference on Information Fusion*. IEEE.

Guerrero, J. L., Berlanga, A., García, J., & Molina, J. M. (2010a). Piecewise linear representation segmentation as a multiobjective optimization problem. In *Advances in Intelligent and Soft Computing*, vol. 79. Springer.

Guerrero, J. L., Berlanga, A., & Molina, J. (2011a). Integrating multicamera surveillance systems into multiagent location systems. In *Advances in Intelligent and Soft Computing*, vol. 94, (pp. 1–9). ACM.

Guerrero, J. L., Berlanga, A., & Molina, J. (2011b). A robust memetic algorithm with self-stopping capabilities. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, (pp. 113–114). ACM.

Guerrero, J. L., Berlanga, A., & Molina, J. (2012a). Fitness-aware Operators for Evolutionary Polygonal Approximation. In *Proceedings of the IADIS International Conference Applied Computing*, (pp. 283–290). IADIS.

Guerrero, J. L., Berlanga, A., & Molina, J. M. (2012b). Initialization procedures for multiobjective evolutionary approaches to the segmentation issue. In *Lecture Notes in Computer Science*, vol. 7208, (pp. 452–463). Springer.

Guerrero, J. L., Berlanga, A., & Molina, J. M. (2013a). An Alternative Archiving Technique for Evolutionary Polygonal Approximation. In *Proceedings of the 5th International Conference on Future Computational Technologies and Applications*, (pp. 68–73).

Guerrero, J. L., Berlanga, A., & Molina, J. M. (2013b). Multiobjective Local Search Techniques for Evolutionary Polygonal Approximation. In *Advances in Intelligent Systems and Computing*, vol. 217, (pp. 171–178). Springer.

Guerrero, J. L., & García, J. (2008). Domain transformation for uniform motion identification in air traffic trajectories. In J. M. Corchado, S. Rodríguez, J. Llinas, & J. M. Molina (Eds.) *Advances in Soft Computing*, vol. 50, (pp. 403–409). Springer.

Guerrero, J. L., & García, J. M., J. amd Molina (2009). An introduction to fusion systems with multi-sensor navigation. In *Proceedings of the 3rd International Workshop on User Centric Technologies and Applications*, (pp. 77–91). Springer.

Guerrero, J. L., Garcia, J., Marti, L., Molina, J., & Berlanga, A. (2009a). A stopping criterion based on kalman estimation techniques with several progress indicators. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, (pp. 587–594). ACM.

Guerrero, J. L., García, J., & Molina, J. M. (2009b). Multi-agent data fusion architecture proposal for obtaining an integrated navigated solution on uav's. In *Lecture Notes in Computer Science*, vol. 5518, (pp. 13–20). Springer.

Guerrero, J. L., Garcia, J., & Molina, J. M. (2010b). Air Traffic Control: A Local Approach to the Trajectory Segmentation Issue. In *Proceedings for the 23rd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, part III Lecture Notes in Artificial Intelligence*, vol. 6098, (pp. 498–507). Springer.

Guerrero, J. L., Gómez-Jordana, A., Berlanga, A., & Molina, J. (2012c). Mutagenesis as a diversity enhancer and preserver in evolution strategies. In *Advances in Intelligent and Soft Computing*, vol. 151, (pp. 725–732). Springer.

Guerrero, J. L., Martí, L., Berlanga, A., Garcia, J., & Molina, J. M. (2010c). Introducing a robust and efficient stopping criterion for moeas. In *Proceedings of the Congress on Evolutionary Computation*, (pp. 1–8). IEEE.

Rebollo, T., Guerrero, J. L., García, J., & Molina, J. M. (2009). A middleware layer for versatile multisensor management under player / stage architecture. In *Proceedings of the 3rd International Workshop on User Centric Technologies and Applications*, (pp. 153–163). Springer.

References

Affenzeller, M., & Winkler, S. (2009). *Genetic algorithms and genetic programming: modern concepts and practical applications*. Chapman & Hall/CRC.

Agrawal, R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. *Foundations of Data Organization and Algorithms*, (pp. 69–84).

Ansari, N., & Huang, K. (1991). Non-parametric dominant point detection. *Pattern Recognition*, *24*(9), 849–862.

Appel, U., & Brandt, A. V. (1983). Adaptive sequential segmentation of piecewise stationary time series. *Information Science*, *29*(1), 27–56.

Arioli, M., Duff, I., & Ruiz, D. (1992). Stopping criteria for iterative solvers. *SIAM Journal on Matrix Analysis and Applications*, *13*, 138.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review*, *61*(3), 183–193.

Babu, B., & Angira, R. (2003). New strategies of differential evolution for optimization of extraction process. In *Proceedings of the International Symposium & 56th Annual Session of IIChE (CHEMCON-2003)*.

Bader, J., & Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, *19*(1), 45–76.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the 2nd International Conference on Genetic Algorithms on Genetic algorithms and their Applications*, (pp. 14–21). L. Erlbaum Associates Inc.

Balzanella, A., Irpino, A., & Verde, R. (2010). Dimensionality Reduction Techniques for Streaming Time Series: A New Symbolic Approach. *Classification as a Tool for Research*, (pp. 381–389).

Baud, O., Gomord, P., Honoré, N., Ostorero, L., Taupin, O., & Tubery, P. (2009). Multi sensor data fusion architectures for air traffic control applications. In N. Milisavljevic (Ed.) *Sensor and Data Fusion*. I-Tech Education and Publishing.

Bäck, T. (1993). Optimal Mutation Rates in Genetic Search. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (pp. 2–8). Morgan Kaufmann Publishers Inc.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, USA.

Bäck, T., Fogel, D., & Michalewicz, Z. (2000). *Evolutionary computation 1: Basic algorithms and operators*. Taylor & Francis.

Bäck, T., & Hoffmeister, F. (1991). Extended selection mechanisms in genetic algorithms. In *Proceedings of the 4th International Conference on Genetic Algorithms*, (pp. 92–99).

Belur, S. (1997). CORE: Constrained optimization by random evolution. In *Late Breaking Papers at the Genetic Programming Conference*, (pp. 280–286).

Bharucha-Reid, A. (1960). *Elements of the theory of Markov processes and their applications*. McGraw-Hill Companies.

Blickle, T., & Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. Tech. Rep. 11, Computer Engineering and Networks Laboratory (TIK).

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. Oxford University Press, USA.

Bramlette, M. F. (1991). Initialization, mutation and selection methods in genetic algorithms for function optimization. In *Proceedings of the 4th International Conference on Genetic Algorithms*, (pp. 100–108). Morgan Kauffman.

Brockwell, P., & Davis, R. (2009). *Time series: theory and methods*. Springer Verlag.

Brown, R. (1996). *Global Positioning: Theory and Applications*, vol. 2, chap. Receiver Autonomous Integrity Monitoring, (pp. 143–165). AIAAA, 2nd ed.

Broyden, C. (1970). The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, *6*(1), 76.

Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., & Schulenburg, S. (2003). Hyper-heuristics: An emerging direction in modern search technology. *Handbook of metaheuristics*, (pp. 457–474).

Burke, E., Newall, J., & Weare, R. (1998). Initialization strategies and diversity in evolutionary timetabling. *Evolutionary computation*, *6*(1), 81–103.

Cantu-Paz, E. (2000). *Efficient and accurate parallel genetic algorithms*. Springer Netherlands.

Cárdenas-Montes, M., Vega-Rodríguez, M. A., & Gómez-Iglesias, A. (2012). Real-world problem for checking the sensitiveness of evolutionary algorithms to the choice of the random number generator. In *Lecture Notes in Computer Science*, vol. 7208, (pp. 385–396). Springer.

Chakraborty, U., & Janikow, C. (2003). An analysis of gray versus binary encoding in genetic search. *Information Sciences*, *156*(3-4), 253–269.

Chipperfield, A., Fleming, P., Pohlheim, H., & Fonseca, C. (1994). Genetic algorithm toolbox user's guide. Tech. Rep. 512, University of Sheffield.

Clifford, G., Azuaje, F., & McSharry, P. (2006). *Advanced methods and tools for ECG data analysis*. Artech House.

Coello, C. (2000). An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)*, *32*(2), 143.

Coello, C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, *191*(11-12), 1245–1287.

Coello, C., & Lamont, G. (2004). *Applications of multi-objective evolutionary algorithms*. World Scientific Pub Co Inc.

Coello, C., Lamont, G., & Van Veldhuizen, D. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer-Verlag New York Inc.

Coleman, T., & Li, Y. (1996). An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on Optimization*, *6*, 418–445.

Corder, G., & Foreman, D. (2009). *Nonparametric statistics for non-statisticians: a step-by-step approach*. John Wiley & Sons Inc.

Corne, D., Knowles, J., & Oates, M. (2000). The pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature PPSN VI*, (pp. 839–848). Springer.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova*, *4*(5), 613–617.

Cronin, T. (1999). A boundary concavity code to support dominant point detection. *Pattern Recognition Letters*, *20*(6), 617–634.

Darwin, C. (1859). On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life. *New York: D Appleton*.

Davis, L. (1985). Job shop scheduling with genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms and their Applications*, (pp. 136–140). L. Erlbaum Associates Inc.

Davis, L. D., & Mitchell, M. (1991). Handbook of genetic algorithms. *Van Nostrand Reinhold*.

De Garis, H. (1990). Genetic programming: Building artificial nervous systems with genetically programmed neural network modules. In *Proceedings of the 7th International Conference on Machine Learning*, (pp. 132–139). Morgan Kauffman.

De Jong, K. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. University of Michigan Ann Arbor, MI, USA.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.

Deb, K., & Agrawal, R. (1995). Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, *9*, 115–148.

Deb, K., Anand, A., & Joshi, D. (2002a). A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary computation*, *10*(4), 371–395.

Deb, K., & Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, *26*, 30–45.

Deb, K., & Jain, S. (2002). Running performance metrics for evolutionary multi-objective optimization. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, (pp. 13–20).

Deb, K., Lele, S., & Datta, R. (2007). A hybrid evolutionary multi-objective and sqp based procedure for constrained optimization. *Advances in Computation and Intelligence*, (pp. 36–45).

Deb, K., Mohan, M., & Mishra, S. (2003). A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. Tech. Rep. 2003002, Indian Institute of Technology Kanpur. Genetic Algorithms Laboratory.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002b). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002c). Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation*, (pp. 825–830). IEEE.

Diggle, P. (1983). *Statistical analysis of spatial point patterns*. Academic Press London.

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, *10*(2), 112–122.

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley.

Dudek, G. (2006). Genetic algorithm with integer representation of unit start-up and shut-down times for the unit commitment problem. *European Transactions on Electrical Power*, *17*(5), 500–511.

Dudoit, S., & van der Laan, M. (2008). *Multiple testing procedures with applications to genomics*. Springer Verlag.

Dunham, J. (1986). Optimum uniform piecewise linear approximation of planar curves. *IEEE transactions on pattern analysis and machine intelligence*, *8*(1), 67–75.

Durillo, J. J., & Nebro, A. J. (2011). jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, *42*(11), 760–771.

Edgeworth, F. (1881). Mathematical Psychics. *History of Economic Thought Books*.

Ehrgott, M. (2005). *Multicriteria optimization*. Springer Verlag.

Ehrgott, M., & Tenfelde-Podehl, D. (2003). Computation of ideal and nadir values and implications for their use in mcdm methods. *European Journal of Operational Research*, *151*(1), 119–139.

Eshelman, L., & Schaffer, J. (1991). Preventing premature convergence in genetic algorithms by preventing incest. In *Proceedings of the 4th International Conference on Genetic Algorithms*, vol. 115, (p. 122). Morgan Kaufmann Publishers.

Espinoza, F. (2003). *A Self-adaptive Hybrid Genetic Algorihm for Optimal Groundwater Remediation Design*. Ph.D. thesis, University of Illinois at Urbana-Champaign.

Espinoza, F., Minsker, B., & Goldberg, D. (2001). Optimal settings for a hybrid genetic algorithm applied to a groundwater remediation problem. In *Proceedings of the World Water and Environmental Resources Congress*.

Farrell, J., & Barth, M. (1999). *The global positioning system and inertial navigation*. McGraw-Hill Professional.

Fleetwood, K. (1999). An introduction to differential evolution. *New ideas in optimization*, (pp. 79–108).

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, *13*(3), 317.

Fogel, L. (1963). Toward inductive inference automata. In *Proceedings of the International Federation for Information Processing Congress*, (p. 395). North-Holland Pub. Co.

Fonseca, C., Da Fonseca, V., & Paquete, L. (2005). Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In *Evolutionary Multi-Criterion Optimization*, (pp. 250–264). Springer.

Fonseca, C., & Fleming, P. (1995). Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, (pp. 45–52).

Fonseca, C., & Fleming, P. (1996). On the performance assessment and comparison of stochastic multiobjective optimizers. In *Parallel Problem Solving from Nature PPSN IV*, (pp. 584–593). Springer.

Fonseca, C., & Fleming, P. (1997). *Handbook of Evolutionary Computation*, chap. Multiobjective Optimization. Institute of Physics Publishing and Oxford University Press, New York.

Fonseca, C., Fleming, P., et al. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (pp. 416–423).

Fourman, M. (1985). Compaction of symbolic layout using genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, (pp. 141–153). L. Erlbaum Associates Inc.

Freeman, H. (1961). On the encoding of arbitrary geometric configurations. *Electronic Computers, IRE Transactions on*, *10*(2), 260–268.

García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization. *Journal of Heuristics*, *15*(6), 617–644.

Garcia, J., Besada, J., Soto, A., & de Miguel, G. (2009). Opportunity trajectory reconstruction techniques for evaluation of ATC systems. *International Journal of Microwave and Wireless Technologies*, *1*(03), 231–238.

Ge, X., & Smyth, P. (2001). Segmental Semi-Markov models for endpoint detection in plasma etching. *IEEE Transactions on Semiconductor Engineering*.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. Wiley-Interscience.

Gionis, A., & Mannila, H. (2005). Segmentation algorithms for time series and sequence data. Tutorial on SIAM International Conference in Data Mining.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, *8*(1), 156–166.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549.

Goldberg, D., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, *1*, 69–93.

Goldberg, D., & Lingle Jr, R. (1985). Alleles, Loci, and the Travelling Salesman Problem. In *Proceedings of the 1st International Conference on Genetic Algorithms and their Applications*, (pp. 154–159). Lawrence Erlbaum Associates.

Goldberg, D., et al. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park.

Goldfarb, D. (1970). A family of variable metric updates derived by variational means. *Mathematics of Computing*, *24*(109), 23–26.

Goutsias, J. (1998). Modeling random shapes: An introduction to random closed set theory. Tech. Rep. JHU/ECE 90-12, John Hopkins University. Department of Electrical and Computer Engineering, Image Analysis and Communications Laboratory, Baltimore, MD 21218.

Grefenstette, J. J. (1987). Introducing problem specific knowledge in genetic algorithms. In *Genetic Algorithms and Simulated Annealing*, (pp. 42–60). Morgan Kauffman.

Groen, F., & Verbeek, P. (1978). Freeman-code probabilities of object boundary quantized contours. *Computer Graphics and Image Processing*, *7*(3), 391–402.

Groves, P. (2008). *Principles of GNSS, inertial, and multi-sensor integrated navigation systems*. Artech House Publishers.

Grunert da Fonseca, V., Fonseca, C., & Hall, A. (2001). Inferential performance assessment of stochastic optimisers and the attainment function. In *Evolutionary Multi-Criterion Optimization*, (pp. 213–225). Springer.

Hanne, T. (1999). On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, *117*(3), 553–564.

Hansen, M., & Jaszkiewicz, A. (1998). Evaluating the quality of approximations to the non-dominated set. Tech. Rep. IMM-REP-1998-7, Technical University of Denmark. Institute of Mathematical Modeling.

Hansen, N., & Kern, S. (2004). Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature PPSN VIII*, (pp. 282–291). Springer.

Hansen, N., Müller, S., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, *11*(1), 1–18.

Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, (pp. 312–317).

Hansen, N., Ostermeier, A., & Gawelczyk, A. (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *Proceedings of the 6th International Conference on Genetic Algorithms*, (pp. 57–64).

Heckbert, P., & Garland, M. (1997). Survey of polygonal surface simplification algorithms. Multiresolution Surface Modeling Course. ACM Siggraph Course notes.

Hedar, A., & Fukushima, M. (2006). Evolution strategies learned with automatic termination criteria. In *Proceedings of the 3rd International Conference on Soft Computing and Intelligent Systems and the 7th International Symposium on Advanced Intelligent Systems*, (pp. 1126–1134).

Hegarty, C. (2006). *Understanding GPS: Principles and Applications*, chap. Least-Squares and Weighted Least-Squares Estmates, (pp. 663–669). MA: Artech House, 2nd ed.

Henderson, C. (1975). Best linear unbiased estimation and prediction under a selection model. *Biometrics*, *31*(2), 423–447.

Hernandez, G., Wilder, K., Nino, F., & Garcia, J. (2005). Towards a self-stopping evolutionary algorithm using coupling from the past. In *Proceedings of the Conference on Genetic and evolutionary computation*, (pp. 615–620). ACM.

Hochba, D. (1997). Approximation algorithms for NP-hard problems. *ACM SIGACT News*, *28*(2), 40–52.

Holland, J. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.

Hollander, M., & Wolfe, D. (1999). *Nonparametric statistical methods*. Wiley-Interscience.

Homaifar, A., Qi, C., & Lai, S. (1994). Constrained optimization via genetic algorithms. *Simulation*, *62*(4), 242.

Hoos, H., & Stützle, T. (2005). *Stochastic local search: Foundations and applications*. Morgan Kaufmann.

Hoos, H. H., & Stutzle, T. (1998). Evaluating las vegas algorithms: Pitfalls and remedies. In *Uncertainty in Artificial Intelligence*, (pp. 238–245).

Horn, J. (1997). *Handbook of evolutionary computation*, chap. Multicriterion decision making. Oxford University Press.

Horn, J., Nafpliotis, N., & Goldberg, D. (1993). Multiobjective optimization using the niched pareto genetic algorithm. Tech. Rep. 93005, University of Illinois. Illinois Genetic Algorithms Laboratory.

Horn, J., Nafpliotis, N., & Goldberg, D. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, vol. 1, (pp. 82–87).

Huband, S., Barone, L., While, L., & Hingston, P. (2005). A scalable multi-objective test problem toolkit. In *Evolutionary Multi-Criterion Optimization*, (pp. 280–295). Springer.

Huntbach, M., & Ringwood, G. (1999). *Agent-oriented programming: from Prolog to guarded definite clauses*. Springer Verlag.

Ishibuchi, H., & Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, *28*(3).

Joines, J., & Houck, C. (1994). On the use of non-stationary penalty functions to solve nonlinearconstrained optimization problems with GA's. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, (pp. 579–584).

Justel, A., Peña, D., & Zamar, R. (1997). A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & probability letters*, *35*(3), 251–259.

Kallel, L., & Schoenauer, M. (1997). Alternative random initialization in genetic algorithms. In *Proceedings of the 7th International Conference on Genetic Algorithms*, (pp. 268–275).

Kalman, R., et al. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35–45.

Kelley, C. (1999). *Iterative methods for optimization*. Society for Industrial Mathematics.

Kelley, C. (2000). Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, *10*(1), 43–55.

Kennedy, J. (2006). Swarm intelligence. *Handbook of Nature-Inspired and Innovative Computing*, (pp. 187–219).

Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, *3*(3), 263–286.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2003). Segmenting time series: A survey and novel approach. *Data mining in time series databases*, (pp. 1–21).

Keogh, E., & Pazzani, M. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, (pp. 239–241). AAAI Press.

Knowles, J. (2002). *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. Ph.D. thesis, Department of Computer Science, University of Reading.

Knowles, J. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *10*(1), 50–66.

Knowles, J., & Corne, D. (2003). Properties of an adaptive archiving algorithm for storing nondominated vectors. *Evolutionary Computation, IEEE Transactions on*, *7*(2), 100–116.

Knowles, J., Thiele, L., & Zitzler, E. (2005). A tutorial on the performance assessment of stochastic multiobjective optimizers. Tech. Rep. 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Kolesnikov, A. (2012). Ise-bounded polygonal approximation of digital curves. *Pattern Recognition Letters*, *33*(10), 1329–1337.

Koza, J. (1992). *Genetic programming: on the programming of computers by means of natural selection*. The MIT press.

Krasnogor, N., & Smith, J. (2005). A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *Evolutionary Computation, IEEE Transactions on*, *9*(5), 474–488.

Laguna, M., & Marti, R. (2003). *Scatter search: methodology and implementations in C*. Springer Netherlands.

Langridge, D. (1972). On the computation of shape. *Frontiers of Pattern Recognition*, (pp. 347–366).

Lanzi, P. (2009). Learning classifier systems. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, (pp. 2853–2878). ACM.

Laumanns, M. (2003). *Analysis and applications of evolutionary multiobjective optimization algorithms*. Ph.D. thesis, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, *10*(3), 263–282.

Letavec, C., & Ruggiero, J. (2002). The n-queens problem. *Informs Transactions on Education*, *2*(3), 101–103.

Liggins, M., Hall, D., & Llinas, J. (2008). *Handbook of multisensor data fusion: theory and practice*. CRC Press.

Liu, X., Lin, Z., & Wang, H. (2008). Novel online methods for time series segmentation. *IEEE Transactions on Knowledge and Data Engineering*, *20*(12), 1616–1626.

Loncaric, S., & Dhawan, A. (1995). Near-optimal MST-based shape description using genetic algorithm. *Pattern Recognition*, *28*(4), 571–579.

Lozano, M., Herrera, F., & Cano, J. (2008). Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information Sciences*, *178*(23), 4421–4433.

Lu, H., & Yen, G. (2003). Rank-density-based multiobjective genetic algorithm and benchmark test function study. *IEEE Transactions on Evolutionary Computation*, *7*(4), 325–343.

Maaranen, H., Miettinen, K., & Mäkelä, M. (2004). Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications*, *47*(12), 1885–1895.

Maaranen, H., Miettinen, K., & Penttinen, A. (2007). On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, *37*(3), 405–436.

Machos, M., Gunopulos, D., & Das, G. (2004). Indexing time-series under conditions of noise. *Data mining in time series databases*, (pp. 67–100).

Maresky, J., Davidor, Y., Gitler, D., Aharoni, G., & Barak, A. (1995). Selectively destructive re-start. In *Proceedings of the 6th International Conference on Genetic Algorithms*, (pp. 144–150). Morgan Kauffman.

Marji, M., & Siy, P. (2003). A new algorithm for dominant points detection and polygonization of digital curves. *Pattern recognition*, *36*(10), 2239–2251.

Martí, L., García, J., Berlanga, A., & Molina, J. (2007). A cumulative evidential stopping criterion for multiobjective optimization evolutionary algorithms. In *Proceedings of the Conference Companion on Genetic and Evolutionary Computation*, (pp. 2835–2842). ACM.

Martí, L., García, J., Berlanga, A., & Molina, J. (2009). An approach to stopping criteria for multi-objective optimization evolutionary algorithms: the mgbm criterion. In *Proceedings of the Congress on Evolutionary Computation*, (pp. 1263–1270). IEEE.

McKay, M., Beckman, R., & Conover, W. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, *21*(2), 239–245.

Meyer, P. (1970). *Introductory Probability and Statistical Applications*. Addison Wesley.

Mühlenbein, H., & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm I. Continuous parameter optimization. *Evolutionary Computation*, *1*(1), 25–49.

Michalewicz, Z. (1995). A survey of constraint handling techniques in evolutionary computation methods. *Evolutionary programming*, *4*, 135–155.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.

Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Boston: Kluwer Academic Publisher.

Mörchen, F. (2003). Time series feature extraction for data mining using dwt and dft. Tech. Rep. 33, Departement of Mathematics and Computer Science Philipps-University Marburg.

Moscato, P. (2000). *Handbook of Applied Optimization*, chap. Memetic Algorithms. Oxford University Press.

Muhlenbein, H. (1994). The breeder genetic algorithm-a provable optimal search algorithm and its application. In *IEE Colloquium on Applications of Genetic Algorithms*, (pp. 5/1–5/3).

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*(4), 308–313.

Oliver, K. (2010). A Review of Constraint-Handling Techniques for Evolution Strategies. *Applied Computational Intelligence and Soft Computing*, *2010*.

Ono, I., Kita, H., & Kobayashi, S. (2003). A real-coded genetic algorithm using the unimodal normal distribution crossover. *Advances in Evolutionary Computing: Theory and Applications*, (pp. 213–237).

Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). Step-size adaptation based on non-local use of selection information. *Parallel Problem Solving from Nature PPSN III*, (pp. 189–198).

Osyczka, A. (1985). Multicriteria optimization for engineering design. *Design Optimization*, (pp. 193–227).

Pal, N., Kundu, M., & Nandi, S. (1998). Application of a new genetic operator in feature selection problems. In *IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, vol. 1, (pp. 37–40). IEEE.

Pareto, V. (1896). Cours d'Economie Politique, volume I and II. *F Rouge, Lausanne*, *250*.

Park, S., Lee, D., & Chu, W. (1999). Fast retrieval of similar subsequences in long sequence databases. In *Proceedings of the 3rd IEEE Knowledge and Data Engineering Exchange Workshop*, (pp. 60–67).

Pavlidis, T., & Ali, F. (2007). Computer recognition of handwritten numerals by polygonal approximations. *Systems, Man and Cybernetics, IEEE Transactions on*, *5*(6), 610–614.

Percival, D., & Walden, A. (2006). *Wavelet methods for time series analysis*. Cambridge Universty Press.

Pohlheim, H. (1995). Ein genetischer Algorithmus mit Mehrfachpopulationen zur Numerischen Optimierung. *at-Automatisierungstechnik 3*, (pp. 127–135).

Rahnamayan, S., Tizhoosh, H., & Salama, M. (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, *53*(10), 1605–1614.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, *1*, 244–256.

Ramsey, C., & Grefenstette, J. (1993). Case-based initialization of genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (pp. 84–91).

Rao, S. (1987). Game theory approach for multiobjective structural optimization. *Computers & Structures*, *25*(1), 119–127.

Ray, B., & Ray, K. (1992). Detection of significant points and polygonal approximation of digitized curves. *Pattern Recognition Letters*, *13*(6), 443–452.

Rechenberg, I., & Eigen, M. (1973). *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog Stuttgart.

Reeves, C. (2003). Genetic algorithms. *Handbook of metaheuristics*, (pp. 55–83).

Rosenberg, B. (1972). The analysis of convex blobs. *Computer Graphics and Image Processing*, *1*(2), 183–192.

Rosenfeld, A., & Johnston, E. (1973). Angle detection on digital curves. *IEEE Transactions on Computers*, *22*(9), 875–878.

Roudenko, O., & Schoenauer, M. (2004). A steady performance stopping criterion for pareto-based evolutionary algorithms. In *Proceedings of the 6th International Multi-Objective Programming and Goal Programming Conference*.

Rudolph, G. (2001). Self-adaptive mutations may lead to premature convergence. *IEEE Transactions on Evolutionary Computation*, *5*(4), 410–414.

Rudolph, G., & Agapie, A. (2000). Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation*, vol. 2, (pp. 1010–1016). IEEE.

Rudolph, G., Naujoks, B., & Preuss, M. (2007). Capabilities of emoa to detect and preserve equivalent pareto subsets. In *Evolutionary Multi-Criterion Optimization*, (pp. 36–50). Springer.

Russell, S., Norvig, P., Canny, J., Malik, J., & Edwards, D. (1995). *Artificial intelligence: a modern approach*. Prentice hall Englewood Cliffs, NJ.

Safe, M., Carballido, J., Ponzoni, I., & Brignole, N. (2004). On stopping criteria for genetic algorithms. In *Advances in Artificial Intelligence–SBIA 2004*, (pp. 405–413). Springer.

Sarfraz, M. (2008). Linear capture of digital curves. In *Interactive Curve Modeling*, (pp. 241–265). Springer London.

Sarfraz, M., Asim, M., & Masood, A. (2004). Piecewise polygonal approximation of digital curves. In *Proceedings of the 8th International Conference on Information Visualisation*, (pp. 991–996). IEEE.

Sarkar, D. (1993). A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition Letters*, *14*(12), 959–964.

Sato, Y. (1992). Piecewise linear approximation of plane curves by perimeter optimization. *Pattern Recognition*, *25*(12), 1535–1543.

Schaffer, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, (pp. 93–100). L. Erlbaum Associates Inc.

Schoenauer, M. (1996). Shape representations and evolution schemes. In *Proceedings of the 5th Annual Conference on Evolutionary Programming*, (pp. 121–129).

Schoenauer, M., & Xanthakis, S. (1993). Constrained ga optimization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (pp. 573–580).

Schultz, A., & Grefenstette, J. (1990). Improving tactical plans with genetic algorithms. In *Proceedings of the 2nd International Conference on Tools for Artificial Intelligence*, (pp. 328–334). IEEE.

Schwefel, H. (1977). *Numerische optimierung von computer-modellen mittels der evolutionsstrategie*. Birkhäuser Basel, Stuttgart.

Schwefel, H. (1981). *Numerical optimization of computer models*. John Wiley & Sons, Inc. New York, NY, USA.

Schwefel, H. (1993). *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc. New York, NY, USA.

Sette, S., Boullart, L., Van Langenhove, L., & Kiekens, P. (1997). Optimizing the fiber-to-yarn production process with a combined neural network/genetic algorithm approach. *Textile research journal*, *67*(2), 84–92.

Shanno, D. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, *24*(111), 647–656.

Shaw, K., Nortcliffe, A., Thompson, M., Love, J., Fleming, P., & Fonseca, C. (1999). Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In *Proceedings of the Congress on Evolutionary Computation*, vol. 1, (pp. 37–45). IEEE.

Sheskin, D. (2000). *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall.

Shin, K., & Lee, Y. (2002). A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications*, *23*(3), 321–328.

Singh, M., Chatterjee, A., & Chaudhury, S. (1997). Matching structural shape descriptions using genetic algorithms. *Pattern Recognition*, *30*(9), 1451–1462.

Sklansky, J., & Gonzalez, V. (1980). Fast polygonal approximation of digitized curves. *Pattern Recognition*, *12*(5), 327–331.

Skolnik, M. (2008). *Radar Handbook*, (third ed.). McGraw-Hill.

Spears, W., & Anand, V. (1991). A study of crossover operators in genetic programming. *Methodologies for Intelligent Systems*, (pp. 409–418).

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), 221–248.

Stoyan, D., Kendall, W., & Mecke, J. (1995). *Stochastic geometry and its applications* . Wiley Series in Probability and Mathematical Statistics.

Stroustrup, B. (1997). *The C++ programming language*. Addison-Wesley Reading, MA.

Surry, P., Radcliffe, N., & Boyd, I. (1995). A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. *Evolutionary Computing*, (pp. 166–180).

Talbi, E. (2009). *Metaheuristics: from design to implementation*. Wiley.

Taylor, S. (2008). *Modelling financial time series*. World Scientific Pub Co Inc.

Teh, C., & Chin, R. (2002). On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(8), 859–872.

Toffolo, A., & Benini, E. (2003). Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation*, *11*(2), 151–167.

Trautmann, H., Wagner, T., Naujoks, B., Preuss, M., & Mehnen, J. (2009). Statistical methods for convergence detection of multi-objective evolutionary algorithms. *Evolutionary computation*, *17*(4), 493–509.

Tsai, Y. (2006). Fast Polygonal Approximation Based on Genetic Algorithms. In *Proceedings of the 5th International Conference on Computer and Information Science and 1st International Workshop on Component-Based Software Engineering, Software Architecture and Reuse*, (pp. 322–326). IEEE.

Tsutsui, S., Yamamura, M., & Higuchi, T. (1999). Multi-parent recombination with simplex crossover in real coded genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, (pp. 657–664).

Ulungu, E., Teghem, J., Fortemps, P., & Tuyttens, D. (1999). MOSA method: a tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, *8*(4), 221–236.

Van Den Bergh, F. (2006). *An analysis of particle swarm optimizers*. Ph.D. thesis, University of Pretoria.

Van-Roy, P., & Haridi, S. (2004). *Concepts, techniques, and models of computer programming*. The MIT Press.

Van Veldhuizen, D. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Ph.D. thesis, Air Force Institute of Technology Wright Patterson AFB, OH, USA.

Van Veldhuizen, D., & Lamont, G. (1998). Multiobjective evolutionary algorithm research: A history and analysis. Tech. Rep. TR-98-03, Air Force Institute of Technology.

Vazirani, V. (2001). *Approximation algorithms*. Springer Verlag.

Vullings, H., Verhaegen, M., & Verbruggen, H. (1997). ECG segmentation using time-warping. In *Lecture Notes in Computer Science*, vol. 1280, (pp. 275–285). Springer.

Wagner, T., Trautmann, H., & Naujoks, B. (2009). Ocd: Online convergence detection for evolutionary multi-objective algorithms based on statistical testing. In *Evolutionary Multi-Criterion Optimization*, (pp. 198–215). Springer.

Wanner, E., Guimaraes, F., Takahashi, R., & Fleming, P. (2006). A quadratic approximation-based local search procedure for multiobjective genetic algorithms. In *Proceedings of the Congress on Evolutionary Computation*, (pp. 938–945). IEEE.

Welch, G., & Bishop, G. (1995). An introduction to the kalman filter. Tech. Rep. TR 95-041, University of North Carolina. Department of Computer Science.

West, M. M., & McCluskey, T. L. (2001). The application of machine learning tools to the validation of an air traffic control domain theory. *International Journal on Artificial Intelligence Tools*, *10*(4), 613–637.

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, *10*(1), 29–38.

Whitley, D., et al. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, vol. 1, (pp. 116–121). Morgan Kaufmann.

Wiese, K., & Glen, E. (2003). A permutation-based genetic algorithm for the RNA folding problem: a critical look at selection strategies, crossover operators, and representation issues. *Biosystems*, *72*(1-2), 29–41.

Williams, G. (2009). GPS for the Sky: a survey of Automatic Dependent Surveillance-Broadcast (ADS-B) and its implementation in the United States. *Journal of Air Law and Commerce*, *74*, 473–701.

Wu, W. (2003). An adaptive method for detecting dominant points. *Pattern Recognition*, *36*(10), 2231–2237.

Yang, Y., Xu, J., & Soh, C. (2006). An evolutionary programming algorithm for continuous global optimization. *European Journal of Operational Research*, *168*(2), 354–369.

Yeang, C.-H., & Jaakkola, T. (2005). Time series analysis of gene expression and location data. *International Journal on Artificial Intelligence Tools*, *14*(5), 755–770.

Yin, P. (1999). Genetic algorithms for polygonal approximation of digital curves. *International journal of pattern recognition and artificial intelligence*, *13*(7), 1061–1082.

Yin, P. Y. (1998). A new method for polygonal approximation using genetic algorithms. *Pattern Recognition Letters*, *19*(11), 1017–1026.

Yu, G., & Schwartz, Z. (2006). Forecasting short time-series tourism demand with artificial intelligence models. *Journal of Travel Research*, *45*(2), 194.

Zaharie, D., & Petcu, D. (2005). Parallel implementation of multi-population differential evolution. In *Proceedings of the NATO Advanced Research Workshop on Concurrent Information Processing and Computing*, (pp. 223–232).

Zhu, Y., Wu, D., & Li, S. (2010). A Piecewise Linear Representation Method of Time Series Based on Feature Points. In *Knowledge-Based Intelligent Information and Engineering Systems*, (pp. 1066–1072). Springer.

Zielinski, K., Peters, D., & Laur, R. (2005). Stopping criteria for single-objective optimization. In *Proceedings of the 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems*.

Zitzler, E., Brockhoff, D., & Thiele, L. (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization*, (pp. 862–876). Springer.

Zitzler, E., Knowles, J., & Thiele, L. (2008). Quality assessment of pareto set approximations. *Multiobjective Optimization*, (pp. 373–404).

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature PPSN VIII*, (pp. 832–842). Springer.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Proceedings of the Conference on Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*.

Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C., & da Fonseca, V. (2002). Why quality assessment of multiobjective optimizers is difficult. In *Proceedings of the Genetic and Evolutionary Computation Conference*, (pp. 666–674).

Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms. a comparative case study. In *Parallel Problem Solving from Nature PPSN V*, (pp. 292–301). Springer.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, *3*(4), 257.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., & da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132.